**ĐẠI HỌC ĐÀ NẴNG**

**TRƯỜNG ĐẠI HỌC KINH TẾ**

🙣 🕮 🙡

****

**BÁO CÁO CHUYÊN ĐỀ**

**THỰC TẬP TỐT NGHIỆP**

**XÂY DỰNG WEBSITE TIẾP NHẬN PHẢN ÁNH TRƯỜNG ĐẠI HỌC KINH TẾ - ĐẠI HỌC**

**ĐÀ NẴNG**

SVTH: Nguyễn Quang Thành

Nguyễn Quốc Thiện

Lớp: 44k21.1

GVHD: ThS. Cao Thị Nhâm

***Đà Nẵng, 06/2022***

LỜI CẢM ƠN

Để hoàn thành báo cáo thực tập này trước hết em xin gửi đến quý thầy, cô giáo trong khoa Thống Kê – Tin Học trường Đại học Kinh Tế Đà Nẵng lời cảm ơn chân thành.

Đặc biệt, em xin gởi đến cô Cao Thị Nhâm người đã tận tình hướng dẫn, giúp đỡ em hoàn thành chuyên đề báo cáo thực tập này lời cảm ơn sâu sắc nhất.

Em xin chân thành cảm ơn Ban Lãnh Đạo, các phòng ban, các anh Mentors và đặc biệt là anh Hiếu giám đốc của Công ty cổ phần BYS, đã tạo điều kiện thuận lợi cho em được tìm hiểu thực tiễn và giúp đỡ em rất nhiều trong suốt quá trình thực tập với công ty.

Đồng thời nhà trường đã tạo cho em có cơ hội được thực tập nơi mà em yêu thích, cho em bước ra đời sống thực tế để áp dụng những kiến thức mà các thầy cô giáo đã giảng dạy. Qua công việc thực tập này bọn em tiếp thu được nhiều kiến thức rất hữu ích để có thể thực hiện tốt công việc lập trình web sau này.

Vì kiến thức bản thân còn hạn chế, trong quá trình thực tập, hoàn thiện chuyên đề này em không tránh khỏi những sai sót, kính mong nhận được những ý kiến đóng góp từ cô cũng như quý công ty.

**Sinh viên**

**Nguyễn Quang Thành và Nguyễn Quốc Thiện**

LỜI CAM ĐOAN

Em xin cam đoan đề tài “*Xây dựng website tiếp nhận phản ảnh Trường Đại học Kinh Tế - Đại học Đà Nẵng”* là một công trình nghiên cứu độc lập của 2 thành viên là Nguyễn Quốc Thiện và Nguyễn Quang Thành dưới sự hướng dẫn của các anh chị Mentors của Công ty cổ phần BYS. Ngoài ra không có bất cứ sự sao chép của người khác. Đề tài, nội dung báo cáo thực tập là sản phẩm mà chúng em đã nỗ lực nghiên cứu trong quá trình học tập tại trường cũng như tham gia thực tập tại công ty. Sản phẩm và kết quả trình bày trong báo cáo là hoàn toàn trung thực, chúng em xin chịu hoàn toàn trách nhiệm, kỷ luật của bộ môn và nhà trường đề ra nếu như có vấn đề xảy ra.

*Sinh viên thực hiện*

*Nguyễn Quang Thành và Nguyễn Quốc Thiện*

# BẢNG PHÂN CHIA CÔNG VIỆC

|  |  |  |
| --- | --- | --- |
| **STT** | **Tasks** | **Assigns to** |
| 1 | Xác định mục tiêu | Thiện |
| 2 | Xác định phạm vi | Thành |
| 3 | Xác định công việc | Thiện |
| 4 | Phân tích nhu cầu | Thiện + Thành |
| 5 | Chia công việc | Thiện + Thành |
| 6 | Xây dựng Usecase | Thiện + Thành |
| 7 | Xây dựng WorkFlow | Thiện + Thành |
| 8 | Thiết kế cơ sở dữ liệu | Thiện |
| 9 | Xây dựng cơ sở dữ liệu | Thiện |
| 10 | Xác định danh sách các API cần thiết | Thiện + Thành |
| 11 | Xây dựng Mock up | Thành |
| 12 | Thiết kế giao diện | Thành |
| 13 | Xây dựng hệ thống phần Back - end | Thiện |
| 14 | Xây dựng hệ thống phần Front - end | Thành |
| 15 | Kiểm thử hệ thống | Thiện + Thành |
| 16 | Fix bugs | Thiện + Thành |
| 17 | Viết báo cáo | Thiện + Thành |
| 18 | Làm slide | Thiện + Thành |

MỤC LỤC

[Chương 1. MỞ ĐẦU 1](#_Toc105072224)

[1.1 SỰ CẦN THIẾT CỦA ĐỀ TÀI 1](#_Toc105072225)

[1.2 MỤC TIÊU NGHIÊN CỨU 2](#_Toc105072226)

[1.3 PHƯƠNG PHÁP THỰC HIỆN 2](#_Toc105072227)

[1.4 BỐ CỤC CỦA BÁO CÁO 3](#_Toc105072228)

[Chương 2. GIỚI THIỆU CÔNG TY CỔ PHẦN BYS 4](#_Toc105072229)

[Chương 3. CƠ SỞ LÝ THUYẾT 6](#_Toc105072230)

[3.1 Tổng quan về website 6](#_Toc105072231)

[3.1.1 Khái niệm website 6](#_Toc105072232)

[3.1.2 Các thành phần và cách thức hoạt động của website 6](#_Toc105072233)

[3.2 Giới thiệu về HTML, CSS, Javascript 7](#_Toc105072234)

[3.2.1 HTML: 7](#_Toc105072235)

[3.2.2 CSS: 9](#_Toc105072236)

[3.2.3 Javascript: 10](#_Toc105072237)

[3.3 ASP.NET CORE MVC 11](#_Toc105072238)

[3.3.1 ASP.NET CORE: 11](#_Toc105072239)

[3.3.2 MVC 12](#_Toc105072240)

[3.3.3 Xây dựng web UI và web API sử dụng ASP.NET Core MVC .12](#_Toc105072241)

[Chương 4. GIỚI THIỆU VỀ CÔNG CỤ PHÁT TRIỂN HỆ THỐNG 14](#_Toc105072242)

[4.1 Visual studio 2019 14](#_Toc105072243)

[4.2 Công cụ quản trị cơ sở dữ liệu Sql Server 2019 Management Studio 16](#_Toc105072244)

[4.2.1 Giới thiệu về SQL – Hệ quản trị cơ sở dữ liệu 16](#_Toc105072245)

[4.3 Github 18](#_Toc105072246)

[4.3.1 Giới thiệu về Github 18](#_Toc105072247)

[4.3.2 Tính năng của Github 19](#_Toc105072248)

[4.3.3 Một số khái niệm cơ bản của Github 19](#_Toc105072249)

[4.4 Visual Studio Code: 20](#_Toc105072250)

[Chương 5. PHÂN TÍCH & THIẾT KẾ WEBSITE TIẾP NHẬN PHẢN ÁNH TRƯỜNG ĐẠI HỌC KINH TẾ- ĐẠI HỌC ĐÀ NẴNG 22](#_Toc105072251)

[5.1 Tổng quan về hệ thống 22](#_Toc105072252)

[5.1.1 Phân tích hệ thống 24](#_Toc105072253)

[5.1.2 Sơ đồ nghiệp vụ 24](#_Toc105072254)

[5.1.3 Sơ đồ Use case tổng quát: 24](#_Toc105072255)

[5.1.4 Mô tả các use case. 28](#_Toc105072256)

[5.1.5 Yêu cầu phi chức năng 56](#_Toc105072257)

[5.2 Thiết kế hệ thống 56](#_Toc105072258)

[5.2.1 Thiết kế kiến trúc hệ thống 56](#_Toc105072259)

[5.2.2 Thiết kế cơ sở dữ liệu 58](#_Toc105072260)

[5.2.3 Thiết kế chức năng 67](#_Toc105072261)

[Chương 6. PHÁT TRIỂN HỆ THỐNG 73](#_Toc105072262)

[6.1 Môi trường phát triển 73](#_Toc105072263)

[6.1.1 Môi trường phát triển của Front-end 73](#_Toc105072264)

[6.1.2 Môi trường phát triển của Back-end 73](#_Toc105072265)

[6.2 Thiết kế giao diện 74](#_Toc105072266)

[6.2.1 Sơ đồ màn hình 74](#_Toc105072267)

[6.2.2 Một số giao diện của hệ thống 74](#_Toc105072268)

[Chương 7. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 85](#_Toc105072269)

[7.1 Kết luận 85](#_Toc105072270)

[7.1.1 Đánh giá tổng quan kì thực tập 85](#_Toc105072271)

[7.1.2 Đánh giá về project trong kì thực tập 85](#_Toc105072272)

[7.1.3 Experience trong kì thực tập 86](#_Toc105072273)

[7.2 Hướng phát triễn sau dự án 86](#_Toc105072274)

[7.2.1 Hướng phát triển của bản thân 86](#_Toc105072275)

[7.2.2 Hướng phát triển của dự án 87](#_Toc105072276)

DANH SÁCH HÌNH ẢNH

[Hình 1. Công ty BYS 4](#_Toc105060618)

[Hình 2. Cách hoạt động của website 7](#_Toc105060619)

[Hình 3. HTML 7](#_Toc105060620)

[Hình 4. CSS 9](#_Toc105060621)

[Hình 5. Javascript 10](#_Toc105060622)

[Hình 6. ASP.NET Core 11](#_Toc105060623)

[Hình 7. Mô hình MVC 12](#_Toc105060624)

[Hình 8. VISUAL STUDIO 2019 14](#_Toc105060625)

[Hình 9. Ngôn ngữ C# 15](#_Toc105060626)

[Hình 10 Kho lưu trữ an toàn 16](#_Toc105060627)

[Hình 11. SQL Server Management Studio 17](#_Toc105060628)

[Hình 12. Github 18](#_Toc105060629)

[Hình 13. Visual studio code 20](#_Toc105060630)

[Hình 14. Workflow 24](#_Toc105060631)

[Hình 15. Sơ đồ Usecase 25](#_Toc105060632)

[Hình 16 Mô hình MVC 57](#_Toc105060633)

[Hình 17. Tổng quan cơ sở dữ liệu 58](#_Toc105060634)

[Hình 18 Chức năng gửi bài đăng chờ kiểm duyệt 68](#_Toc105060635)

[Hình 19 Chức năng tạo bản nháp 68](#_Toc105060636)

[Hình 20 Chức năng sửa bản nháp 69](#_Toc105060637)

[Hình 21 Chức năng xoá bản nháp 69](#_Toc105060638)

[Hình 22 Chức năng xem danh sách bài đăng và xem bài đăng chi tiết 70](#_Toc105060639)

[Hình 23 Xem danh sách bài đăng chờ duyệt, phản hồi và duyệt publish 70](#_Toc105060640)

[Hình 24 Chức năng xem thống kê bài đăng tổng quát 71](#_Toc105060641)

[Hình 25 Chức năng gửi mail cảnh báo 72](#_Toc105060642)

[Hình 26 Sơ đồ màn hình 74](#_Toc105060643)

[Hình 27. Giao diện trang đăng nhập 74](#_Toc105060644)

[Hình 28. Giao diện trang chủ 75](#_Toc105060645)

[Hình 29. Giao diện trang thay đổi mật khẩu 75](#_Toc105060646)

[Hình 30. Giao diện trang xem bài đăng chi tiết 76](#_Toc105060647)

[Hình 31. Giao diện trang xem bài đăng tổng quan cá nhân 76](#_Toc105060648)

[Hình 32. Giao diện trang tạo bài đăng mới 77](#_Toc105060649)

[Hình 33. Giao diện trang duyệt bài đăng tổng quan( Nhân viên) 77](#_Toc105060650)

[Hình 34. Giao điện trang duyệt bài đăng chi tiết ( Nhân viên) 78](#_Toc105060651)

[Hình 35. Giao diện trang Access Denied 78](#_Toc105060652)

[Hình 36. Giao diện trang chủ ( Quản trị viện) 79](#_Toc105060653)

[Hình 37. Giao diện trang tạo tải khoản sinh viên 79](#_Toc105060654)

[Hình 38. Giao diện trang tạo tải khoản cho phòng ban 80](#_Toc105060655)

[Hình 39. Giao diện trang tạo tài khoản ban giám hiệu 80](#_Toc105060656)

[Hình 40. Giao diện trang quản lý tài khoản người dùng 81](#_Toc105060657)

[Hình 41. Giao diện trang phân quyền 81](#_Toc105060658)

[Hình 42. Giao diện trang tạo mới phòng ban 82](#_Toc105060659)

[Hình 43. Giao diện trang quản lý phòng ban 82](#_Toc105060660)

[Hình 44. Giao diện trang thống kê 83](#_Toc105060661)

DANH SÁCH TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Từ viết tắt** | **Diễn giải** |
| MVC | Model View Controller |
|  |  |
|  |  |
|  |  |
|  |  |

# MỞ ĐẦU

## SỰ CẦN THIẾT CỦA ĐỀ TÀI

Trong những năm gần đây, cùng với sự phát triển của xã hội, công nghệ thông tin nói chung, các trang mạng xã hội nói riêng, đặc biệt là Internet phát triển mạnh hơn bao giờ hết, nó đã và đang ảnh hưởng rất lớn (cả tích cực và tiêu cực) đến mọi hoạt động và sinh hoạt của con người, nhất là giới trẻ. Với đặc điểm nổi trội là tính kết nối nhanh, chia sẻ rộng, chỉ cần một chiếc điện thoại hay một máy tính kết nối Internet, chúng ta có thể truy cập và tham gia vào rất nhiều trang mạng như: Facebook, Zalo, Youtube, Twitter… trong đó, phổ biến nhất là Facebook.

Vấn đề này không chỉ ảnh hưởng đến đời sống sinh hoạt thường nhật của con người mà còn ảnh hưởng đến các môi trường khác như môi trường giáo dục, môi trường lao động xã hội, … bất cứ vấn đề nào cũng có thể đưa trên mạng xã hội để bàn tán, thảo luận cho dù có phải là thông tin chính thống hay đã qua xác minh hay không. Nó sẽ gây ra các hệ luỵ khá nghiêm trọng là khiến người đọc bị sai lệch về tư duy, nhận thức nhất là với thế hệ trẻ, thanh thiếu niên, cái thế hệ còn tò mò khi sắp hoặc đang rời xa vòng tay bảo bọc của cha mẹ.

Tại trường Đại Học Kinh Tế Đà Nẵng, bọn em nhận thấy có thực trạng sinh viên hoặc các cá nhân đang làm việc tại trường khi có những thắc mắc, góp ý hay là không hài lòng thậm chí là bức xúc sẽ không chọn phương án lên các phòng ban liên quan để làm việc trực tiếp hoặc sử dụng hòm thư góp ý mà sẽ sử dụng các kênh mạng xã hội như Facebook cụ thể là các trang như “DUE Confession”, “Sinh viên Đại Học Kinh Tế”, … để đăng tải các ý kiến của mình nhằm mục đích được sự ủng hộ của “cư dân mạng” và tìm ra giải pháp, hướng giải quyết cho vấn đề mặc dù các thông tin này vẫn chưa được kiểm chứng hay xác minh làm rõ. Điều này sẽ ảnh hưởng đến uy tín nhà trường nói chung và các phòng ban, giảng viên nói riêng. Mặc dù đã đăng lên các kênh này nhưng nhà trường có thể vẫn sẽ không nắm được các vấn đề để giải quyết vì các kênh này không phải là các kênh làm việc chính thống của nhà trường, các vấn đề vẫn có thể sẽ tiếp tục diễn ra. Sau đó sẽ ảnh hưởng đến mặt truyền thông của nhà trường, tạo ra các hệ luỵ sau này.

Đứng trước thực trạng đó, cần một hệ thống để tiếp nhận và giải quyết, xử lý mà ở bất kì đâu sinh viên cũng có thể nêu ra những ý kiến, phàn nàn, bức xúc của mình một cách tiện lợi và dễ dàng. Môi trường này sẽ kết nối sinh viên, đối tác với nhà trường một cách chặt chẽ và cũng khiến cho nhà trường có thể kiểm soát thông tin, những trải nghiệm của sinh viên một cách minh bạch và rõ ràng, hướng tới một môi trường giáo dục tốt hơn.

Với ý nghĩa đó, tụi em xin chọn đề tài “*Xây dựng website tiếp nhận phản ánh Trường Đại học Kinh Tế - Đại học Đà Nẵng*” làm đề tài thực tập tốt nghiệp của mình.

## MỤC TIÊU NGHIÊN CỨU

Xây dựng một website tiếp nhận phản ánh hiện đại, tiện lợi, thay thế phương thức phản ánh truyền thống nhưng vẫn thân thiện với người dùng, dễ sử dụng, đáp ứng tính minh bạch trong việc xử lý các phản ánh và ý kiến của sinh viên, có đầy đủ các tính năng cho quản trị viên và ban giám hiệu trong hoạt động quản trị hệ thống.

Nâng cao kiến thức về Công nghệ thông tin, kỹ năng lập trình, thu thập thêm kinh nghiệm làm một dự án hoàn chỉnh có đầy đủ các bước. Học hỏi và rút ra nhiều kiến thức từ các anh chị đi trước.

## PHƯƠNG PHÁP THỰC HIỆN

* Nghiên cứu thực trạng và xây dựng quy trình tiếp nhận phản ánh phù hợp để giải quyết tốt nhất vấn đề đang tồn đọng.
* Phân tích thiết kế hệ thống phù hợp với quy trình đã nghiên cứu và giải quyết thực trạng đang có.
* Nghiên cứu, tìm hiểu lý thuyết về HTML, CSS, JavaScript, Bootstrap và Framwork ASP.NET Core để xây dựng nên trang web.
* Nghiên cứu và tìm hiểu kỹ thuật lập trình, hiểu được cách thức hoạt động của Client – Server.
* Thực hiện thiết kế, xây dựng website dưới sự giúp đỡ và chỉ dẫn của các mentor và giảng viên hướng dẫn

## BỐ CỤC CỦA BÁO CÁO

Báo cáo gồm 7 chương:

* **Chương 1: Mở Đầu.**
* **Chương 2: Giới thiệu công ty cổ phần BYS.**
* **Chương 3: Cơ sở lý thuyết.**
* **Chương 4: Giới thiệu công cụ phát triển.**
* **Chương 5: Phân tích và thiết kế website tiếp nhận phản ánh Trường Đại học Kinh Tế- Đà Nẵng.**
* **Chương 6: Phát triển hệ thống.**
* **Chương 7: Kết luận và hướng phát triển.**

# GIỚI THIỆU CÔNG TY CỔ PHẦN BYS



Hình 1. Công ty BYS

Thành lập tại Việt Nam năm 2010, Công ty cổ phần BYS chuyên cung cấp dịch vụ tư vấn triển khai giải pháp Quản lý tổng thể doanh nghiệp (ERP). Được thành lập bởi các chuyên gia nhiều kinh nghiệm trong lĩnh vực Công nghệ thông tin & tư vấn triển khai ERP có nhiều năm làm việc tại Đức và Úc, cộng với sự hỗ trợ về công nghệ, đào tạo và chuyển giao giải pháp từ Nhà cung cấp giải pháp ERP của nước ngoài.

Với lợi thế nhiều năm học tập và làm việc trong lĩnh vực triển khai ERP tại nước ngoài, tự hào với trí tuệ người Việt, công ty tâm huyết xây dựng một giải pháp ERP “đúng nghĩa” đáp ứng được nhu cầu quản lý của đa số doanh nghiệp doanh nghiệp trong & ngoài nước, đủ sức cạnh tranh với giải pháp nước ngoài.

Với khẩu hiệu “Build Your Solution - Build Your Success ”, BYS với đội ngũ Chuyên gia và Cán bộ nhân viên không ngừng nỗ lực, sáng tạo và nâng cao chất lượng giải pháp. Chúng tôi cam kết sẽ đồng hành cùng phát triển với doanh nghiệp theo phương châm: “Thành công của khách hàng làm nên thành công của chúng tôi”.

Với tầm nhìn của công ty là trở thành 1 trong những nhà cung cấp hàng đầu về Giải pháp quản lý tổng thể doanh nghiệp (ERP) tại Việt Nam và các nước trong khu vực.

Với sứ mệnh tôn trọng và cam kết mang đến cho khách hàng giải pháp quản lý tốt nhất với chi phí hợp lý. Đồng thời, xây dựng mối quan hệ tin cậy, uy tín, hợp tác cùng phát triển với các đối tác trong lĩnh vực Công nghệ thông tin.

# CƠ SỞ LÝ THUYẾT

## Tổng quan về website

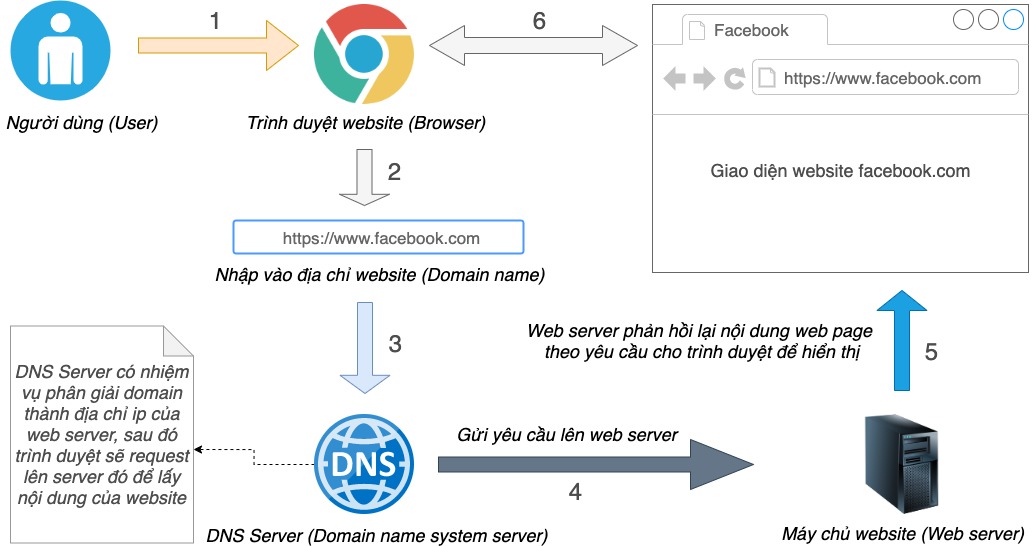
### Khái niệm website

Website là một tập hợp các trang web (web pages) bao gồm văn bản, hình ảnh, video, flash v.v... thường chỉ nằm trong một tên miền (domain name) hoặc tên miền phụ (subdomain) trên World Wide Web của Internet. Trang web được lưu trữ (web hosting) trên máy chủ web (web server) có thể truy cập thông qua Internet.  
  Một trang web là tập tin HTML hoặc XHTML có thể truy nhập dùng giao thức HTTP hoặc HTTPS. Website có thể được xây dựng từ các tệp tin HTML (website tĩnh) hoặc vận hành bằng các CMS chạy trên máy chủ (website động). Website có thể được xây dựng bằng nhiều ngôn ngữ lập trình khác nhau (PHP,.NET, Java, Ruby on Rails...).

### Các thành phần và cách thức hoạt động của website

Cách thức hoạt động của website được mô tả tóm tắt như sau:

1. Trình duyệt Web sẽ thực hiện một truy vấn dựa vào tên miền (domain) để tìm ra địa chỉ IP thực sự của web server tương ứng chứa website có tên miền đó bằng một giao thức đặc biệt gọi là DNS.
2. Sau khi đã tìm được địa chỉ IP, trình duyệt sẽ gửi gói tin yêu cầu – HTTP request đến địa chỉ của web server, yêu cầu trả về nội dung trang web. Gói tin yêu cầu đó cũng như tất cả các gói tin, dữ liệu khác trao đổi giữa máy chủ với máy chúng ta (gọi là máy khách) được thực hiện qua một bộ giao thức TCP/IP.
3. Khi nhận được các yêu cầu từ máy khách, máy chủ web sẽ tiến hành kiểm tra và nếu có thể đáp ứng các yêu cầu đó thì nó sẽ gửi lại cho máy khách các tập tin được yêu cầu. Thông thường máy chủ sẽ trả về tập tin HTML để hiển thị trên trình duyệt, có liên kết đến những tập tin hình ảnh, âm thanh,… khác. Các tập tin này có thể được chia thành nhiều gói tin (packets) nhỏ và gửi về cho trình duyệt của người dùng đang ở máy khách.
4. Khi nhận được, trình duyệt sẽ ghép những gói tin nhỏ nhận được thành những tập tin hoàn chỉnh và hiển thị lên màn hình. Như thế là chúng ta có một trang web hoàn chỉnh để xem.



Hình 2. Cách thức hoạt động của website

## Giới thiệu về HTML, CSS, Javascript

### HTML:



Hình 3. HTML

HTML hay HyperText Markup Language – Ngôn ngữ đánh dấu siêu văn bản, là ngôn ngữ được sử dụng cho các tài liệu web. HTML là một ngôn ngữ xác định đâu là ý nghĩa, mục đích và cấu trúc của một tài liệu. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho các Website.

Một Website thường chứa nhiều trang con và mỗi trang con này lại có một tập tin HTML riêng. Lưu ý, HTML không phải là ngôn ngữ lập trình. Điều này có nghĩa là nó không thể thực hiện các chức năng “động”. Hiểu một cách đơn giản hơn, cũng tương tự như phần mềm Microsoft Word, HTML chỉ có tác dụng bố cục và định dạng trang web. HTMLkhi kết hợp với CSS và JavaScriptsẽ trở thành một nền tảng vững chắc cho thế giới mạng.

HTML document có đuôi file dạng .html hoặc htm. Bạn có thể xem chúng bằng các trình duyệt web hiện hành như Google Chrome, Firefox, Safari,… Nhiệm vụ của trình duyệt là đọc những file HTML này và “biến đổi” chúng thành một dạng nội dung visual trên Internet sao cho người dùng có thể xem và hiểu được chúng.

Thông thường, một Website sẽ có nhiều HTML document (ví dụ: trang chủ, trang blog, trang liên hệ, …) và mỗi trang con như vậy sẽ có một tệp HTML riêng. Mỗi tài liệu HTML bao gồm 1 bộ tag (hay còn gọi là element). Nó tạo ra một cấu trúc tương tự như cây thư mục với các heading, section, paragraph, … và một số khối nội dung khác. Hầu hết tất cả các HTML element đều có một tag mở và một tag đóng với cấu trúc <tag></tag>.

### CSS:



Hình 4. CSS

CSS là viết tắt của cụm từ “Cascading Style Sheets”, tạm dịch: ngôn ngữ tạo

phong cách cho trang web. Ngôn ngữ lập trình này quy định cách các thành phần

HTML của trang web thực sự sẽ xuất hiện trên frontend như thế nào.

CSS được tạo ra để kết hợp với ngôn ngữ markup HTML để tạo phong cách

cho trang web.

Có 3 loại style CSS chính:

* Style CSS Internal: là style được tải lên mỗi khi trang web được refresh.
* Style CSS Inline: bạn có thể chỉnh sửa một yếu tố nào đó mà không cần truy

cập trực tiếp vào file CSS.Trang 7

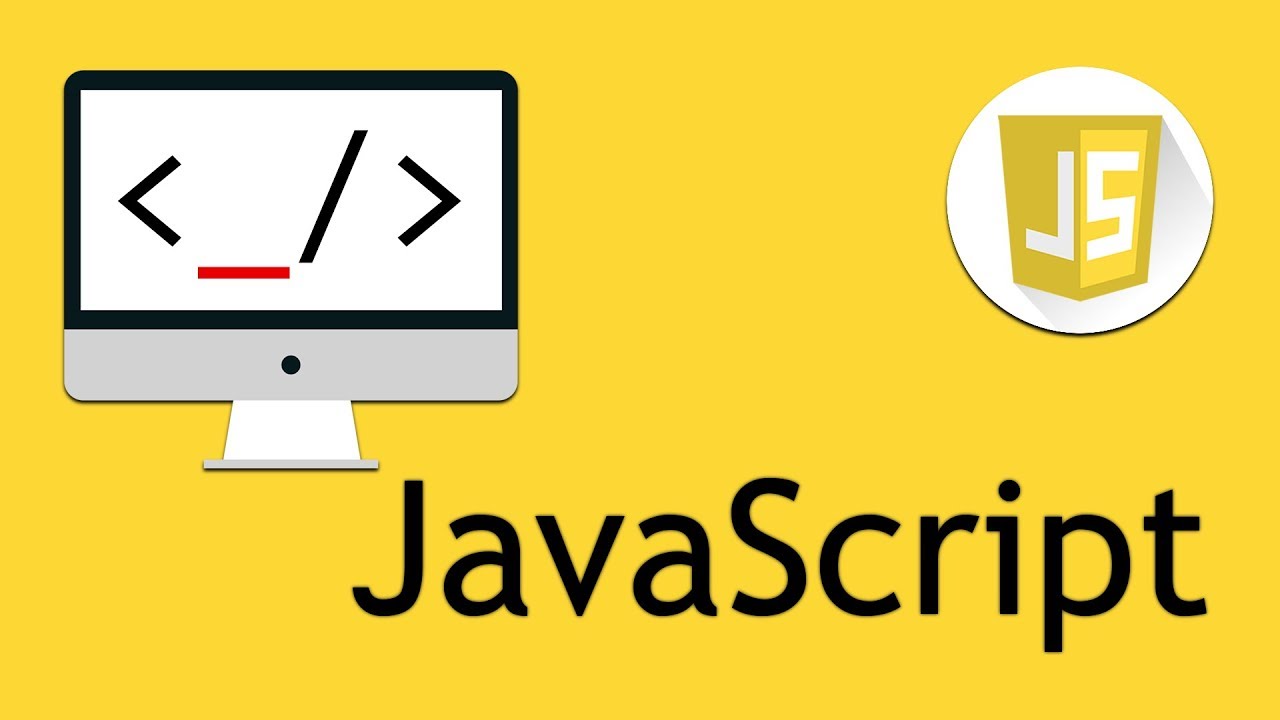
* External style: bạn có thể tạo phong cách ở file khác áp dụng CSS vào trang

bạn muốn. External style sẽ cải thiện thời gian tải trang rất nhiều.

* Đặc biệt, có thể sử dụng External Style CSS để tạo phong cách cho nhiều

trang cùng lúc.

### Javascript:



Hình 5. Javascript

Javascript là một ngôn ngữ lập trình hoặc ngôn ngữ kịch bản cho phép bạn triển khai các tính năng phức tạp trên trang web như hiển thị cập nhật nội dung hiện thời, bản đồ tương tác, hoạt hình 2D / đồ họa 3D,… Ngoài ra còn cho phép bạn tạo nội dung động, kiểm soát đa phương tiện, hình ảnh động và hầu hết mọi thứ khác. Nó thường được tích hợp và nhúng vào trong HTML giúp cho Website trở nên sống động hơn, cho phép kiểm soát các hành vi của trang Web tốt hơn so với khi chỉ sử dụng mỗi HTML. Javascript được hỗ trợ hầu hết trên tất cả các trình duyệt như Firefox, Chrome, Safari,… thậm chí các trình duyệt trên thiết bị di động.

Javascript ngày nay có thể được ứng dụng trong rất nhiều lĩnh vực:

* Ứng dụng trong lập trình Website: Khi nhắc tới lập trình Web thì chắc chắn không thể không nhắc tới bộ 3 HTML, CSS và Javascript. Có thể nói không phải là tất cả, tuy nhiên hầu hết các Website đang chạy hiện nay đều sử dụng đến Javascript hoặc những Framework của nó như: Bootstrap, jQuery Foundation, UIKit,… Javascript giúp tạo nên các hiệu ứng hiển thị trên Website, các tương tác với người dùng.
* Xây dựng các ứng dụng di động, trò chơi và ứng dụng trên desktop: Nếu bạn có hứng thú với phát triển trò chơi trên Internet, bạn có thể cân nhắc đến sử dụng kiến thức Javascript để tạo ra các trò chơi trên trình duyệt. Mặc dù sẽ có những hạn chế xoay quanh độ phức tạp của trò chơi dựa trên trình duyệt Web, song Javascript vẫn có thể được sử dụng tốt như bất kỳ ngôn ngữ nào khác khi nói đến lập trình trò chơi.

## ASP.NET CORE MVC



Hình

Hình 6. ASP.NET Core

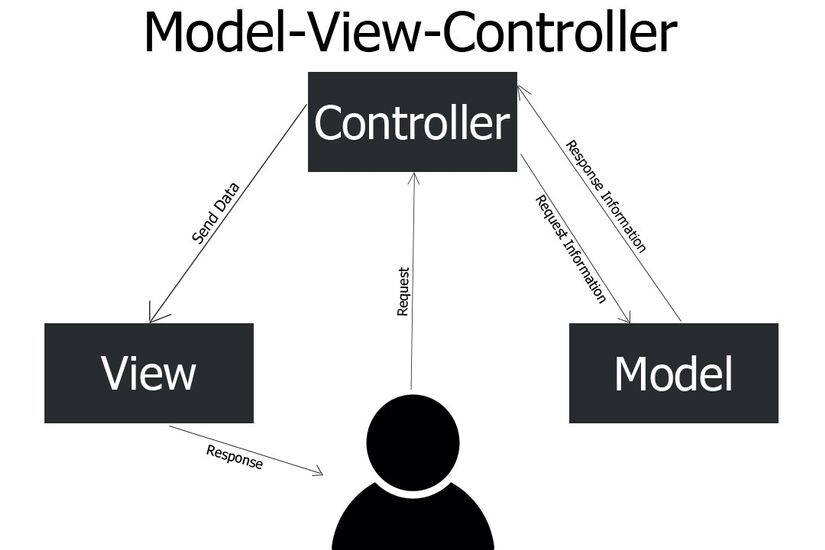
### ASP.NET CORE:

ASP.NET CORE là một open-source mới và framework đa nền tảng (cross-platform) cho việc xây dựng những ứng dụng hiện tại dựa trên kết nối đám mây, giống như web apps, IoT và backend cho mobile.

Ứng dụng ASP.NET Core có thể chạy trên .NET Core hoặc trên phiên bản đầy đủ của .NET Framework. Nó được thiết kế để cung cấp và tối ưu development framework cho những dụng cái mà được triển khai trên đám mây (clound) hoặc chạy on-promise.

Nó bao gồm các thành phần theo hướng module nhằm tối thiểu tài nguyên và chi phí phát triển, như vậy bạn giữ lại được sự mềm giẻo trong việc xây dựng giải pháp của bạn. Bạn có thể phát triển và chạy những ứng dụng ASP.NET Core đa nền tảng trên Windows, Mac và Linux.

### MVC



Hình 7. Mô hình MVC

MVC (viết tắt của Model-View-Controller) là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. MVC chia một ứng dụng thành ba phần tương tác được với nhau để tách biệt giữa cách thức mà thông tin được xử lý nội hàm và phần thông tin được trình bày và tiếp nhận từ phía người dùng. Khi sử dụng đúng cách, mẫu MVC giúp cho người phát triển phần mềm cô lập các nguyên tắc nghiệp vụ và giao diện người dùng một cách rõ ràng hơn. Phần mềm phát triển theo mẫu MVC tạo nhiều thuận lợi cho việc bảo trì vì các nguyên tắc nghề nghiệp và giao diện ít liên quan với nhau. Trong mẫu Model-View-Controller, mô hình (model) tượng trưng cho dữ liệu của chương trình phần mềm. Tầm nhìn hay khung nhìn (view) bao gồm các thành phần của giao diện người dùng. Bộ kiểm tra hay bộ điều chỉnh (controller) quản lý sự trao đổi giữa dữ liệu và các nguyên tắc nghề nghiệp trong các thao tác liên quan đến mô hình.

### Xây dựng web UI và web API sử dụng ASP.NET Core MVC

Có thể tạo ra các ứng dụng web có thể testing theo mô hình MVC(Model-View-Controller) cùng với một số đặc điểm nổi bật:

* Xây dựng HTTP services cái mà hỗ trợ nhiều định dạng và có đầy đủ hỗ trợ cho nội dung của dữ liệu trả về.
* Razor của ASP.NET cung cấp một ngôn ngữ hiệu quả để tạo ra Views.
* Tag Helper cho phép code server side tham gia vào việc tạo và render phần tử HTML
* Model Binding tự động ánh xạ dữ liệu từ HTTP request tới tham số của method action.
* Model Validation tự động thực hiện validate client và serve

Với ASP.NET CORE, chúng ta đạt được một số nền tảng cải tiến:

* Hợp nhất việc xây dựng web UI và web APIs.
* Tích hợp những client-side frameworks hiện đại và những luồng phát triển.
* Hệ thống cấu hình dựa trên môi trường đám mây thật sự.
* Dependency injection được xây dựng sẵn.
* HTTP request được tối ưu nhẹ hơn.
* Có thể host trên IIS hoặc self-host trong process riêng.
* Được xây dựng trên .NET Core, hỗ trợ thực sự app versioning.
* Chuyển các thực thể, thành phần, module như những NuGet packages.
* Những công cụ mới để đơn giản hóa quá trình phát triển web hiện đại.
* Xây dựng và chạy đa nền tảng (Windows, Mac và Linux).
* Mã nguồn mở và tập trung vào cộng đồng.

# GIỚI THIỆU VỀ CÔNG CỤ PHÁT TRIỂN HỆ THỐNG

## Visual studio 2019



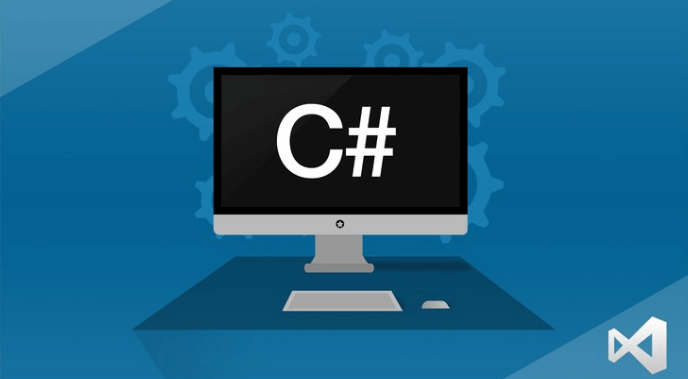
Hình 8. VISUAL STUDIO 2019

Visual studio là một môi trường phát triển tích hợp (IDE) từ Microsoft. Nó được sử dụng để phát triển chương trình máy tính cho Microsoft Windows, cũng như các trang web, các ứng dụng web và các dịch vụ web. Visual Studio sử dụng nền tảng phát triển phần mềm của Microsoft như Windows API, Windows Forms, Windows Presentation Foundation, Windows Storevà Microsoft Silverlight. Nó có thể sản xuất cả hai ngôn ngữ máy và mã số quản lý.

Visual Studio hỗ trợ nhiều ngôn ngữ lập trình khác nhau và cho phép trình biên tập mã và gỡ lỗi để hỗ trợ (mức độ khác nhau) hầu như mọi ngôn ngữ lập trình. Các ngôn ngữ tích hợp gồm có C, C++ và C++/CLI , VB.NET, C# và F#. Hỗ trợ cho các ngôn ngữ khác như J++/J#, Python và Ruby thông qua dịch vụ cài đặt riêng rẽ. Nó cũng hỗ trợ XML/XSLT, HTML /XHTML, JavaScript và CSS.

Các tính năng của Visual studio code:

Đa nền tảng: Phần mềm lập trình Visual Studio của Microsoft hỗ trợ sử dụng trên nhiều nền tảng khác nhau. Không giống như các trình viết code khác, Visual Studio sử dụng được trên cả Windows, Linux và Mac Systems. Điều này cực kỳ tiện lợi cho lập trình viên trong quá trình ứng dụng.



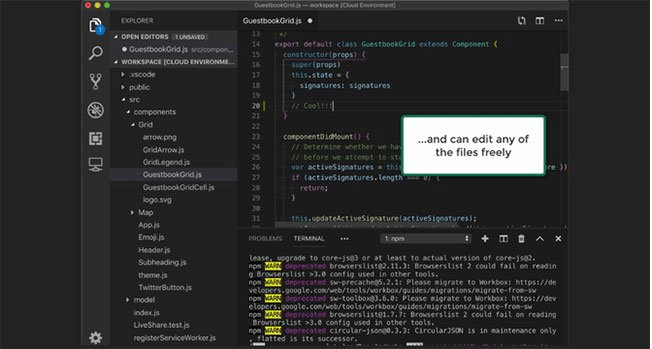
**Hình 9. Ngôn ngữ C#**

Phần mềm hỗ trợ Đa ngôn ngữ lập trình: Không chỉ hỗ trợ đa nền tảng, Visual Studio cũng cho phép sử dụng nhiều ngôn ngữ lập trình khác nhau từ C#, F#, C/C++, HTML, CSS, [Visual Basic](https://vi.wikipedia.org/wiki/Visual_Basic#:~:text=Visual%20Basic%20(vi%E1%BA%BFt%20t%E1%BA%AFt%20VB,h%E1%BB%A3p%20(IDE)%20k%E1%BA%BFt%20b%C3%B3.), JavaScript,… Bởi vậy, Visual Studio có thể dễ dàng phát hiện và thông báo cho bạn khi các chương trình có lỗi.

Kho tiện ích mở rộng phong phú: Mặc dù Visual Studio có hệ thống các ngôn ngữ hỗ trợ lập trình khá đa dạng. Nhưng nếu lập trình viên muốn sử dụng một ngôn ngữ khác, bạn có thể dễ dàng tải xuống các tiện ích mở rộng. Tính năng hấp dẫn này được hoạt động như một phần chương trình độc lập nên không lo làm giảm hiệu năng của phần mềm.

Lưu trữ phân cấp: Phần lớn các tệp dữ liệu đoạn mã của  Visual Studio đều được đặt trong các thư mục tương tự nhau. Đồng thời, Visual Studio cũng cung cấp một số thư một cho các tệp đặc biệt để bạn lưu trữ an toàn, dễ tìm, dễ sử dụng hơn.

Kho lưu trữ an toàn: Với Visual Studio, bạn có thể hoàn toàn yên tâm về tính lưu trữ, bởi phần mềm đã được kết nối GIT và một số kho lưu trữ an toàn được sử dụng phổ biến hiện nay.



Hình 10 Kho lưu trữ an toàn

Hỗ trợ Git: Do kết nối với [GitHub](https://github.com/) nên Visual Studio cho phép hỗ trợ sao chép, kéo thả trực tiếp. Các mã code này sau đó cũng có thể thay đổi và lưu lại trên phần mềm.

## Công cụ quản trị cơ sở dữ liệu Sql Server 2019 Management Studio

### Giới thiệu về SQL – Hệ quản trị cơ sở dữ liệu

Hệ quản trị cơ sở dữ liệu có tên tiếng Anh là Database Management System (DBMS). Nó là một hệ thống phần mềm cho phép tạo lập cơ sở dữ liệu. Đồng thời điều khiển các truy cập đến cơ sở dữ liệu đó.

Mục đích sử dụng hệ quản trị cơ sở dữ liệu:

* Quản lý lượng dữ liệu lớn hiệu quả
* Quản lý những cơ sở dữ liệu tồn tại lâu dài
* Hỗ trợ quan sát dữ liệu
* Hỗ trợ định nghĩa cấu trúc dữ liệu và thao tác trên dữ liệu thông qua các ngôn ngữ cao cấp nhất định
* Cung cấp các truy cập chính xác cùng lúc với nhiều truy cập dữ liệu khác nhau.
* Đảm bảo tính độc lập và bất biến của dữ liệu dù cấu trúc mô hình dữ liệu có thay đổi.
* Bảo vệ dữ liệu khỏi các truy cập không được cấp phép
* Phục hồi dữ liệu, tránh làm mất dữ liệu trên hệ thống



**Hình 11. SQL Server Management Studio**

SQL Server chính là một hệ quản trị dữ liệu quan hệ sử dụng câu lệnh SQL để trao đổi dữ liệu giữa máy cài SQL Server và máy Client. Một Relational Database Management System – RDBMS gồm có: databases, datase engine và các chương trình ứng dụng dùng để quản lý các bộ phận trong RDBMS và những dữ liệu khác.

SQL Server không phải là một hệ quản trị cơ sở dữ liệu độc lập mà nó chỉ là một thành phần với vai trò ngôn ngữ là công cụ giao tiếp giữa hệ cơ sở dữ liệu và người dùng. Chính vì thế nó được sử dụng trong các dịch vụ [thiết kế web đẹp](https://mona.media/thiet-ke-website-tai-hcm/) với chức năng giao tiếp với người dùng với các vai trò sau:

* Đòi hỏi có tính tương tác cao: Người dùng có thể dễ dàng trao đổi với các tiện ích thông qua các câu lệnh của SQL đến cơ sở dữ liệu và nhận kết quả từ cơ sở dữ liệu.
* Là một ngôn ngữ lập trình cơ sở dữ liệu: Các lập trình viên có thể xây dựng các chương trình ứng dụng giao tiếp với cơ sở dữ liệu bằng cách nhúng các câu lệnh SQL vào trong ngôn ngữ lập trình.
* Là một ngôn ngữ lập trình quản trị cơ sở dữ liệu: Người quản trị cơ sở dữ liệu có thề quản lý, định nghĩa và điều khiển truy cập cơ sở dữ liệu thông qua SQL.
* Lập trình cho các hệ thống chủ khách: SQL được sử dụng như là một công cụ giao tiếp với các trình ứng dụng trong hệ thống cơ sở dữ liệu khách chủ.
* Truy cập dữ liệu trên Internet: SQL được sử dụng với vai trò tương tác với dữ liệu trong hầu hết các máy chủ web và máy chủ Internet.
* Cơ sở dữ liệu phân tán: Với vai trò giao tiếp với các hệ thống trên mạng, gửi và nhận các yêu cầu truy xuất dữ liệu với nhau.

Nhìn chung SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ được sử dụng trong các chức năng tương tác giữa người dùng và dữ liệu. Chính vì thế, nó được sử dụng trong các dịch vụ thiết kế web rẻ và chuyên nghiệp.

## Github

### Giới thiệu về Github

GitHub là một hệ thống quản lý dự án và phiên bản code, hoạt động giống như một mạng xã hội cho lập trình viên. Các lập trình viên có thể clone lại mã nguồn từ một repository và Github chính là một dịch vụ máy chủ repository công cộng, mỗi người có thể tạo tài khoản trên đó để tạo ra các kho chứa của riêng mình để có thể làm việc.



Hình 12. Github

Github cung cấp các tính năng social networking như feeds, followers, và network graph để các developer học hỏi kinh nghiệm của nhau thông qua lịch sử commit.

Nếu một comment để mô tả và giải thích một đoạn code. Thì với Github, commit message chính là phần mô tả hành động mà bạn thực hiện trên source code.

### Tính năng của Github

GitHub được coi là một mạng xã hội dành cho lập trình viên lớn nhất và dễ dùng nhất với các tính năng cốt lõi như:

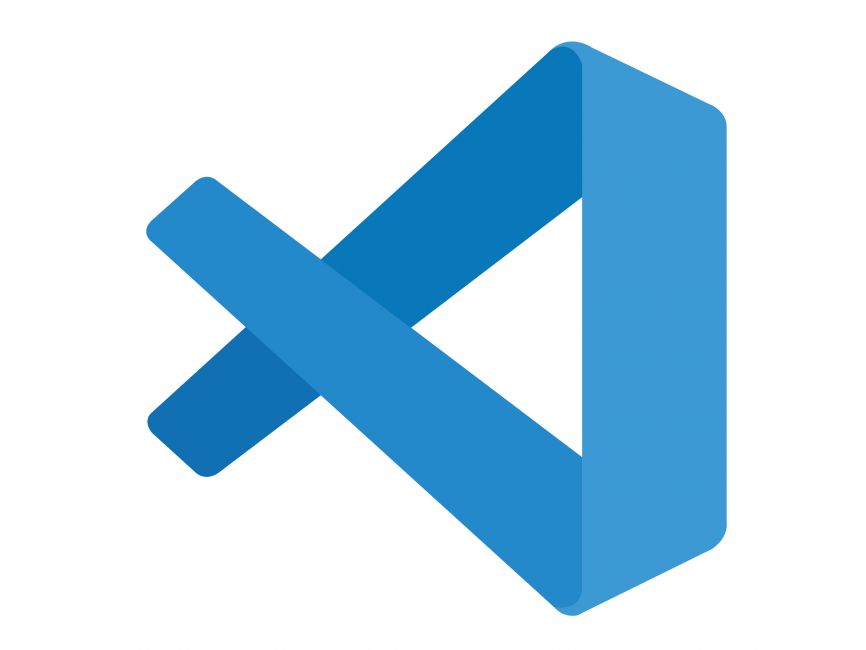
* Wiki, issue, thống kê, đổi tên project, project được đặt vào namespace là user.
* Watch project: theo dõi hoạt động của project của người khác. Xem quá trình người ta phát triển phầm mềm thế nào, project phát triển ra sao.
* Follow user: theo dõi hoạt động của người khác.

Có 2 cách tiếp cận GitHub: Tạo project của riêng mình Contribute cho project có sẵn: fork project có sẵn của người khác, sửa đổi, sau đó đề nghị họ cập nhật sửa đổi của mình (tạo pull request).

### Một số khái niệm cơ bản của Github

* **Git**: là prefix của các lệnh được sử dụng dưới CLI
* **branch**: được hiểu như là nhánh, thể hiện sự phân chia các version khi 2 version đó có sự sai khác nhất định và 2 version đều có sự khác nhau.
* **commit**: là một điểm trên cây công việc (Work Tree ) hay gọi là cây phát triển công việc
* **clone**: được gọi là nhân bản, hay thực hiện nhân bản. Sử dụng để clone các project, repository trên các hệ thống chạy trên cơ sở là git, ví dụ như: bitbucket, github, gitlab, cor(1 sản phẩm mã nguồn mở cho phép người dùng tự tạo git server cho riêng mình trên vps, server),… Việc clone này sẽ sao chép repository tại commit mình mong muốn, dùng để tiếp tục phát triển. Thao tác này sẽ tải toàn bộ mã nguồn, dữ liệu về máy tính của bạn.
* **folk**: Folk là thao tác thực hiện sao chép repository của chủ sở hữu khác về git account của mình. sử dụng và đối xử như 1 repository do mình tạo ra.
* **repository**: Kho quản lý dữ liệu, là nơi lưu trữ các dữ liệu, mã nguồn của project.
* **tag**: sử dụng để đánh dấu một commit khi bạn có quá nhiều commit tới mức không thể kiểm soát được.
* **remote**: sử dụng để điều khiển các nhánh từ một repository trên git server, đối xử với các nhánh trên remote tương tự như đối xử với các nhánh trên local
* **diff**: So sánh sự sai khác giữa phiên bản hiện tại với phiên bản muốn so sánh, nó sẽ thể hiện các sự khác nhau
* **.gitignore**: file mặc định của git sử dụng để loại bỏ (ignore) các thư mục, file mà mình không muốn push lên git server

## Visual Studio Code:



Hình 13. Visual studio code

Visual Studio Code (VS Code hay VSC) là một trong những trình soạn thảo mã nguồn phổ biến nhất được sử dụng bởi các lập trình viên. Nhanh, nhẹ, hỗ trợ đa nền tảng, nhiều tính năng và là mã nguồn mở chính là những ưu điểm vượt trội khiến VS Code ngày càng được ứng dụng rộng rãi.

Có thể nói, Visual Studio Code là sự kết hợp cực kỳ hoàn hảo giữa IDE và Code Editor. Nó hỗ trợ cho người dùng rất nhiều tiện ích như: đổi theme, hỗ trợ Git, cải tiến mã nguồn, có syntax highlighting, hỗ trợ cho quá trình gõ code, sử dụng các phím tắt và nhiều tùy chọn khác nhau,…

Hiện nay, Visual Studio Code chiếm ưu thế ở hầu hết các môi trường phát triển dành cho lập trình viên. Trong một khảo sát của Stack Overflow (năm 2019), Visual Studio Code được đánh giá là môi trường phát triển được dùng phổ biến nhất với hơn 50% lượt bình chọn trong tổng số hơn 90 nghìn người dùng tham gia khảo sát. Trong khi đó, con số này của năm 2018 chỉ có 35%. Điều này cho thấy độ phủ sóng ngày càng rộng rãi và sự hữu ích mà VSCode mang lại cho lập trình viên.

# PHÂN TÍCH & THIẾT KẾ WEBSITE TIẾP NHẬN PHẢN ÁNH TRƯỜNG ĐẠI HỌC KINH TẾ- ĐẠI HỌC ĐÀ NẴNG

## Tổng quan về hệ thống

Mục Tiêu của hệ thống nhằm tạo ra một hệ thống cho sinh viên trường Đại học Kinh Tế- Đại Học Đà nẵng có thể gửi những phản ánh, thắc mắc cho nhà trường giải quyết một cách nhanh chóng và dễ dàng nhất.

Ý nghĩa khi hoàn thành hệ thống sẽ giải quyết được một số thực trạng đang còn tồn đọng trong khẩu tiếp nhận và phản hồi các thắc mắc cũng như phản ánh của sinh viên Đại học Kinh Tế- Đại học Đà Nẵng.

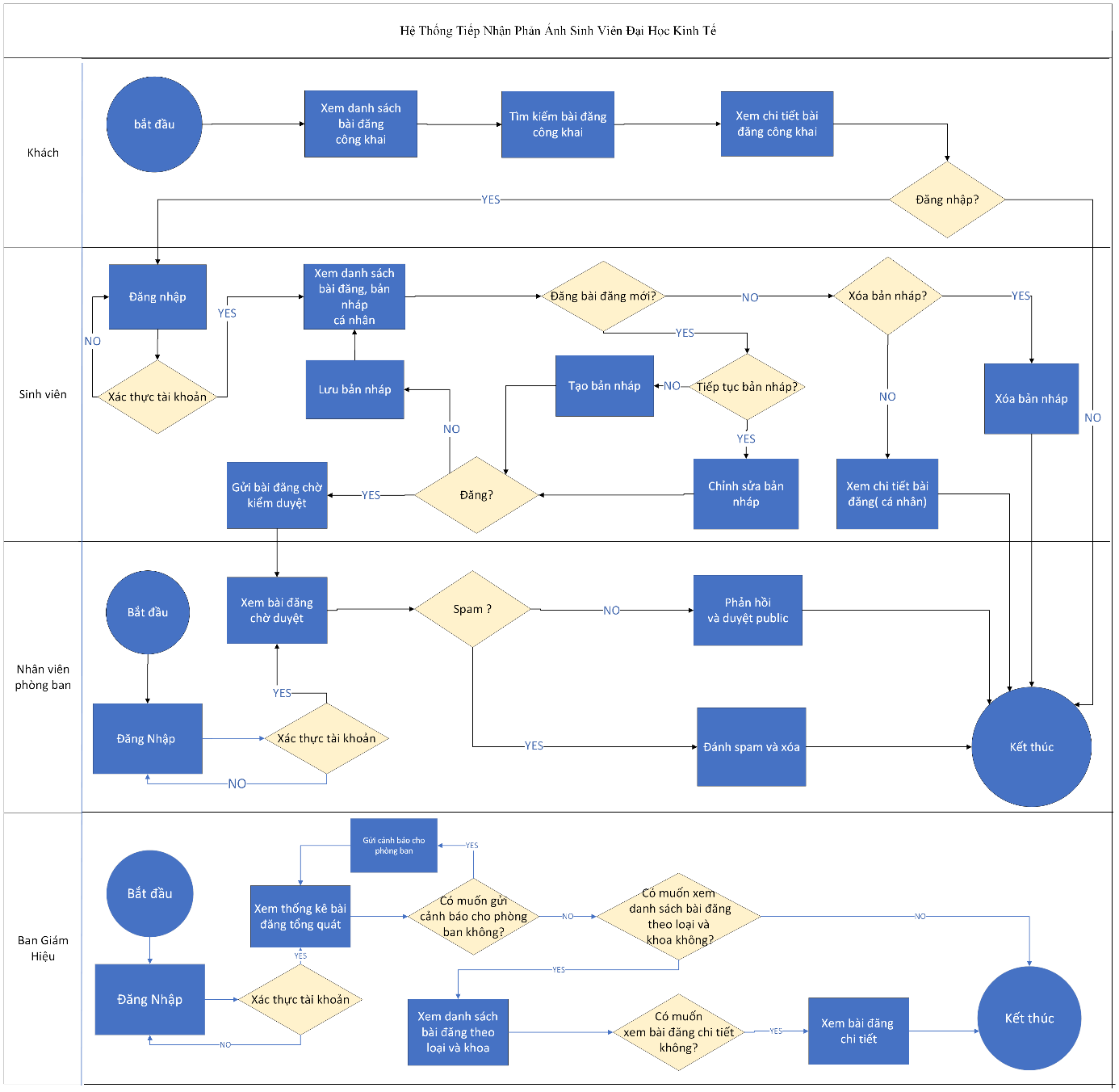
Website “Tiếp nhận phản ánh của sinh viên trường Đại học Kinh Tế- Đại Học Đà Nẵng” được thiết kế để người dùng tiếp cận hệ thống có thể dễ dàng và nhanh chóng nhất.

Hệ thống được thiết kế với:

* Giao diện thân thiện, hài hoà, đơn giãn giúp người dùng có thể dễ dàng sử dụng.
* Trang chủ sẽ hiện thị tổng quan các bài đăng của các sinh viên đã được phòng ban tiếp nhận phản ánh duyệt và có thể dễ dàng tìm kiếm các bài viết thông qua từ khoá ở tiêu đề và nội dung bài đăng.
* Người đăng bài có thể yên tâm về danh tính vì phòng ban tiếp nhận phản ánh không thể biết được tên của người đăng bài.
* Tài khoản của sinh viên được tạo tự động bằng cách nhập tự động từ hệ thống tài khoản DUE có sẵn.
* Hệ thống được thiết kế với 5 tác nhân: Ban giám hiệu, admin, phòng ban tiếp nhận phản ánh, sinh viên, khách. Chức năng và vai trò chính của từng actor trong hệ thống sẽ là:
* Ban giám hiệu: Có vai trò xem thông kê tổng quát và báo cáo tình trạng số liệu của hệ thông theo các khoảng thời gian khác nhau cũng như có thể gửi mail cảnh báo cáo các phòng ban tiếp nhận phản ánh nếu thấy vấn đề.
* Admin: Có vai trò quản lý tài khoản người dùng và quản lý tài khoản phòng ban tiếp nhận phản ánh như tạo mới, cấp mới mật khẩu vô hiệu hoá hoặc khoá tài khoản đó.
* Phòng ban tiếp nhận phản ánh: Có vai trò tiếp nhận những bài đăng mà sinh viên hoặc phụ huynh của sinh viên gửi đến. Sau khi phản hồi bài đăng sau đó sẽ quyết định duyệt bài đăng đó hoặc đánh spam để gửi mail thống báo có người đăng bài về tình trạng của bài đăng.
* Sinh viên: Có thể gửi những phản ánh những ý kiến, tâm tư của mình cho nhà trường để nhận những giải đáp về những thắc mắc đó. Hoặc có thể gửi những xem những bài đăng đã được duyệt do những sinh viên đã gửi trước đây.
* Khách: Có thể xem chi tiết các bài đăng đã được duyệt.

### Phân tích hệ thống

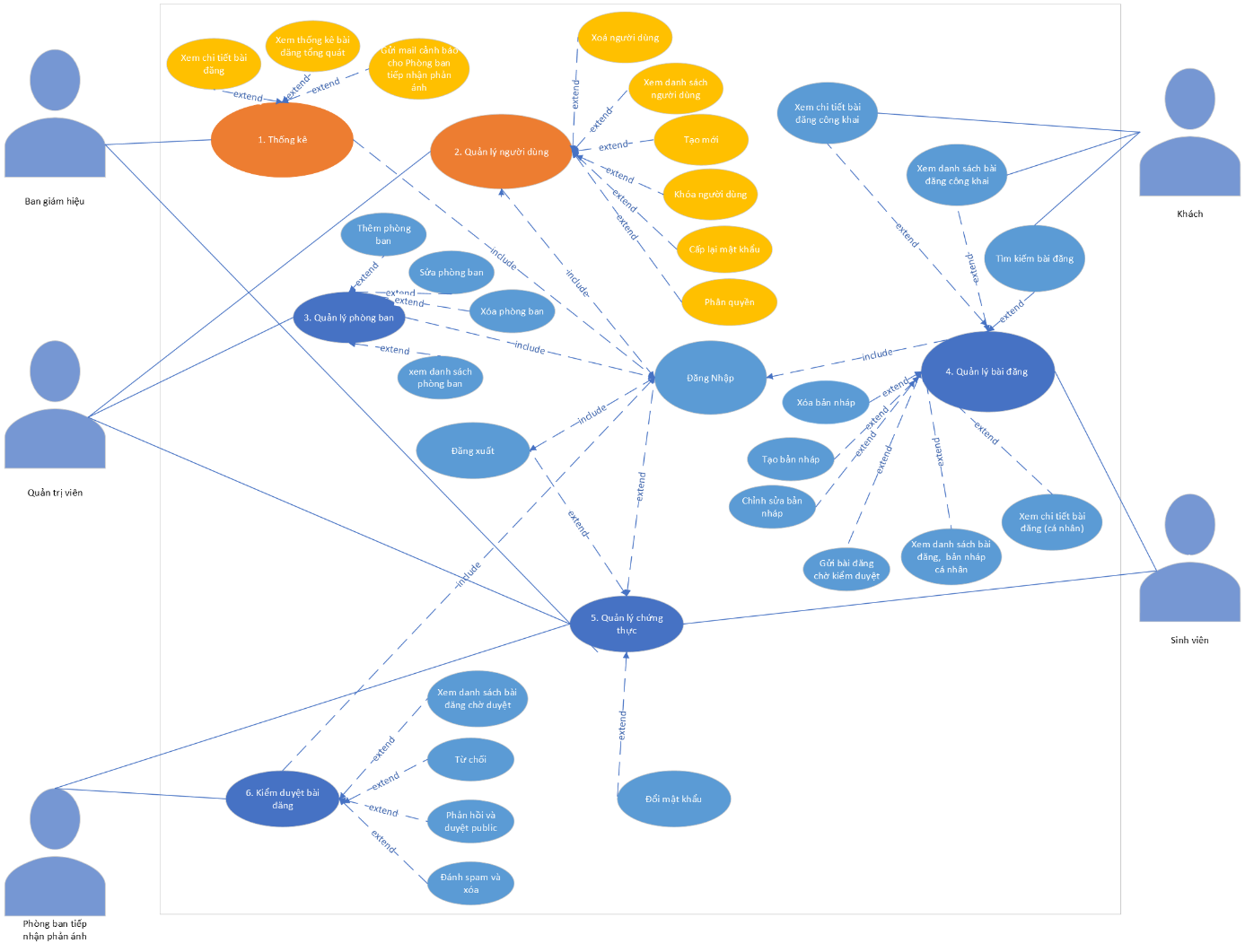
### Sơ đồ nghiệp vụ



Hình 14. Workflow

### Sơ đồ Use case tổng quát:

Hệ thống gồm 5 tác nhân không phân cấp: Khách, Sinh viên, Phòng ban tiếp nhận phản ánh, Ban giám hiệu và Quản trị viên



Hình 15. Sơ đồ Usecase

#### 5.2.1.1 Danh sách các tác nhân và mô tả

|  |  |
| --- | --- |
| Tác nhân | Mô tả tác nhân |
| Khách | Người có vai trò là khách vãng lai, sử dụng hệ thống với các chức năng về bài đăng công khai là: Tìm kiếm, xem danh sách, xem chi tiết. |
| Sinh viên | Người có vai trò là sinh viên thuộc nhà trường, sử dụng hệ thống với chức năng Quản lý bài đăng. |
| Phòng ban tiếp nhận phản ánh | Người có vai trò thuộc các phòng ban của nhà trường, sử dụng hệ thống với chức năng Kiểm duyệt bài đăng. |
| Quản trị viên | Người có vai trò quản trị website về mặt hệ thống, sử dụng hệ thống với chức năng Quản lý phòng ban và Quản lý người dùng. |
| Ban giám hiệu | Người có vai trò là ban giám hiệu thuộc nhà trường, đảm bảo tính minh bạch và giám sát hoạt động phòng ban của hệ thống, sử dụng hệ thống với chức năng Thống Kê. |

#### 5.2.1.2 Danh sách use-case và mô tả

|  |  |
| --- | --- |
| Tên UC | Tên chức năng phân rã |
| Thống kê | Thông kê bài đăng |
| Xem chi tiết bài đăng |
| Gửi mail phản ánh cho phòng ban tiếp nhận phản ánh |
| Quản lý người dùng | Xem danh sách người dùng |
| Tạo mới |
| Khoá người dùng |
| Cấp lại mật khẩu |
| Xoá người dùng |
| Cấp quyền người dùng |
| Quản lý phòng ban | Xem danh sách phòng ban |
| Tạo phòng ban mới |
| Sửa phòng ban |
|  | Xoá phòng ban |
| Quản lý bài đăng | Xem danh sách bài đăng, bản nháp cá nhân |
| Xem bài đăng chi tiết (cá nhân) |
| Tạo bản nháp |
| Chỉnh sửa bản nháp |
| Xoá bản nháp |
| Lưu bản nháp |
| Gửi bài đăng chờ kiểm duyệt |
| Tìm kiếm bài đăng |
| Xem danh sách bài đăng công khai |
| Xem chi tiết bài đăng công khai |
| Quản lý chứng thực | Đăng nhập |
| Đăng xuất |
| Đổi mật khẩu |
| Kiểm duyệt bài đăng | Xem bài đăng chờ duyệt |
| Phản hồi và duyệt public |
| Từ chối |
| Đánh spam và xoá |

### Mô tả các use case.

#### Thống kê

##### Thống kê bài đăng

|  |  |
| --- | --- |
| **Tác nhân:** Ban giám hiệu | **Loại UC:** business use-case |
| **Mô tả tổng quát:**  Usecase liệt kê ra cho phép ban giám hiệu xem thống kê chi tiết số lượng bài đăngtheo loại và theo phòng ban. | |
| **Kích hoạt:** Ban giám hiệu muốn xem thống kê chi tiết số lượng bài đăng theo loại và theo phòng ban | |
| **Mối quan hệ:** Chức năng thống kê bài đăng có mối quan hệ với xem chi tiết bài đăng và gửi mail phản ánh cho phòng ban. | |
| **Luồng xử lý chính:**   1. Ban giám hiệu đăng nhập vào hệ thống 2. Chọn nút “Thống Kê” 3. Hệ thống hiện thị bảng thống kê số lượng bài đăng theo loại và phòng ban theo thời gian thực. (Ban giám hiệu có thể chọn xem thống kê theo ngày, tháng, năm trước đó) | |
| **Luồng xử lý ngoại lệ:** Khi lỗi server/mất mạng xảy ra người dùng không thể truy cập và chỉnh sửa nhật ký, hệ thống hiện thông báo lỗi | |

##### Xem chi tiết bài đăng

|  |  |
| --- | --- |
| **Tác nhân:** Ban giám hiệu | **Loại UC:** business use-case |
| **Mô tả tổng quát:**  Usecase cho phép ban giám hiệu xem chi tiết thông tin các bài đăng. | |
| **Kích hoạt:** Ban giám hiệu muốn xem chi tiết nội dung các bài đăng. | |
| **Mối quan hệ:**   * Người dùng thực hiện chức năng xem chi tiết bài đăng * Xem chi tiết bài đăng là chức năng mở rộng của use case thống kê | |
| **Luồng xử lý chính:**   1. Ban giám hiệu đăng nhập vào hệ thống 2. Chọn nút “Thống Kê” 3. Chọn bài đăng muốn xem. | |
| **Luồng xử lý ngoại lệ:**  Khi lỗi server/mất mạng xảy ra người dùng không thể truy cập và chỉnh sửa nhật ký, hệ thống hiện thông báo lỗi | |

##### Gửi mail cảnh báo cho phòng ban tiếp nhận phản ánh

|  |  |
| --- | --- |
| **Tác nhân:** Ban giám hiệu | **Loại UC:** business use-case |
| **Mô tả tổng quát:**  Usecase cho phép ban giám hiệu gửi mail cảnh báo cho từng phòng ban nếu thấy có sự chậm trễ trong việc xử lý bài đăng. | |
| **Kích hoạt:** Ban giám hiệu muốn gửi mail cảnh báo cho các phòng ban xử lý bài đăng chậm. | |
| **Mối quan hệ:**   * Người dùng thực hiện chức năng xem chi tiết bài đăng * Xem chi tiết bài đăng là chức năng mở rộng của use case thống kê | |
| **Luồng xử lý chính:**   1. Ban giám hiệu đăng nhập vào hệ thống 2. Chọn nút “Thống Kê”. 3. Chọn biểu tượng dấu “!” bên cạnh phòng ban đó, nếu muốn gửi mail cảnh báo cho phòng ban | |
| **Luồng xử lý ngoại lệ:**  Khi lỗi server/mất mạng xảy ra người dùng không thể truy cập và chỉnh sửa nhật ký, hệ thống hiện thông báo lỗi | |

#### Quản lý người dùng

##### Xem danh sách người dùng

|  |  |
| --- | --- |
| **Tác nhân:** Quản trị viên | **Loại UC:** business use-case |
| **Mô tả tổng quát:**  Usecase cho phép quản trị viên xem danh sách các tài khoản đang tồn tại | |
| **Kích hoạt:** Quản trị viên muốn xem danh sách tài khoản đang tồn tại | |
| **Mối quan hệ:**   * Người dùng thực hiện chức năng xem danh sách tài khoản * Xem danh sách tài khoản là chức năng mở rộng của use case quản lý người dùng | |
| **Luồng xử lý chính:**   1. Quản trị viên đăng nhập vào hệ thống và chọn nút “Quản Trị”. 2. Chọn “Xem và sửa tài khoản người dùng” để xem. | |
| **Luồng xử lý ngoại lệ:**  Khi lỗi server/mất mạng xảy ra người dùng không thể truy cập và chỉnh sửa nhật ký, hệ thống hiện thông báo lỗi | |

##### Tạo tài khoản người dùng

|  |  |
| --- | --- |
| **Tác nhân:** Quản trị viên | **Loại UC:** business use-case |
| **Mô tả tổng quát:**  Usecase cho phép quản trị viên tạo mới tài khoản người dùng theo loại, riêng lẻ hoặc theo file excel có sẵn. | |
| **Kích hoạt:** Quản trị viên muốn tạo tài khoản người dùng | |
| **Mối quan hệ:**   * Người dùng thực hiện chức năng tạo tài khoản người dùng * Tạo tài khoản người dùng là chức năng mở rộng của use case quản lý người dùng | |
| **Luồng xử lý chính:**   1. Quản trị viên đăng nhập vào hệ thống chọn “Quản Trị”. 2. Chọn “Tạo tài khoản người dùng”. 3. Chọn loại cần tạo là “sinh viên”, “nhân viên phòng ban” hoặc “quản trị viên”. 4. Điền thông tin và nhấn tạo mới | |
| **Luồng xử lý thay thế:**  4b. Quản trị viên tải excel mẫu về  5.Điền đầy đủ thông tin và đăng tải file excel danh sách tài khoản muốn tạo  6.Nhấn tạo mới để tạo nhiều tài khoản cùng lúc | |
| **Luồng xử lý ngoại lệ:**  4c. Hệ thống xác thực thông tin đăng ký không thành công và hiển thị thông báo.  Use case dừng lại  4c1. Quản trị viên sửa lại thông tin tài khoản cần tạo. Usecase tiếp tục Usecase 4. | |

##### Khoá người dùng

|  |  |
| --- | --- |
| **Tác nhân:** Quản trị viên | **Loại UC:** business use-case |
| **Mô tả tổng quát:**  Usecase cho phép quản trị viên khoá/ mở khoá các tài khoản đang tồn tại | |
| **Kích hoạt:** Quản trị viên muốn khoá/mở khoá tài khoản đang tồn tại | |
| **Mối quan hệ:**   * Người dùng thực hiện chức năng khoá tài khoản * Xem danh sách tài khoản là chức năng mở rộng của use case quản lý người dùng | |
| **Luồng xử lý chính:**   1. Quản trị viên đăng nhập vào hệ thống chọn “Quản Trị”. 2. Chọn “Xem và chỉnh sửa tài khoản”. 3. Chọn biểu tượng ổ khoá nếu muốn khoá/ mở khoá tài khoản. | |
| **Luồng xử lý ngoại lệ:**  Khi lỗi server/mất mạng xảy ra người dùng không thể truy cập và chỉnh sửa nhật ký, hệ thống hiện thông báo lỗi | |

##### Cấp lại mật khẩu

|  |  |
| --- | --- |
| **Tác nhân:** Quản trị viên | **Loại UC:** business use-case |
| **Mô tả tổng quát:**  Usecase cho phép cấp lại mật khẩu cho các tài khoản đang tồn tại khi nhận được yêu cầu | |
| **Kích hoạt:** Quản trị viên muốn cấp lại cho tài khoản đang tồn tại | |
| **Mối quan hệ:**   * Người dùng thực hiện chức năng cấp lại mật khẩu * Cấp lại mật khẩu là chức năng mở rộng của use case quản lý người dùng | |
| **Luồng xử lý chính:**   1. Quản trị viên đăng nhập vào hệ thống chọn “Quản Trị”. 2. Chọn “Xem và chỉnh sửa tài khoản”. 3. Chọn biểu tượng chìa khoá nếu muốn đặt lại mật khẩu tài khoản. 4. Tài khoản được đặt lại mật khẩu mặt định | |
| **Luồng xử lý ngoại lệ:**  Khi lỗi server/mất mạng xảy ra người dùng không thể truy cập và chỉnh sửa nhật ký, hệ thống hiện thông báo lỗi | |

##### Xoá tài khoản

|  |  |
| --- | --- |
| **Tác nhân:** Quản trị viên | **Loại UC:** business use-case |
| **Mô tả tổng quát:**  Usecase cho phép xoá các cho các tài khoản đang tồn tại | |
| **Kích hoạt:** Quản trị viên muốn xoá tài khoản đang tồn tại | |
| **Mối quan hệ:**   * Người dùng thực hiện chức năng xoá tài khoản * Xoá tài khoản là chức năng mở rộng của use case quản lý người dùng | |
| **Luồng xử lý chính:**   * 1. Quản trị viên đăng nhập vào hệ thống chọn “Quản Trị”.   2. Chọn “Xem và chỉnh sửa tài khoản”.   3. Chọn biểu tượng chìa thùng rác nếu muốn xoá tài khoản đó. | |
| **Luồng xử lý ngoại lệ:** Khi lỗi server/mất mạng xảy ra người dùng không thể truy cập và chỉnh sửa nhật ký, hệ thống hiện thông báo lỗi | |

#### Quản lý phòng ban

##### Xem danh sách phòng ban

|  |  |
| --- | --- |
| **Tác nhân:** Quản trị viên | **Loại UC:** business use-case |
| **Mô tả tổng quát:**  Usecase cho phép xoá các cho các tài khoản đang tồn tại | |
| **Kích hoạt:** Quản trị viên muốn xem danh sách phòng ban đang tồn tại. | |
| **Mối quan hệ:**   * Người dùng thực hiện chức năng xem danh sách phòng ban * Xem danh sách phòng ban là chức năng mở rộng của use case quản lý phòng ban | |
| **Luồng xử lý chính:**   1. Quản trị viên đăng nhập vào hệ thống chọn “Quản Trị”. 2. Chọn “Xem và chỉnh sửa phòng ban” để xem. | |
| **Luồng xử lý ngoại lệ:**  Khi lỗi server/mất mạng xảy ra người dùng không thể truy cập và chỉnh sửa nhật ký, hệ thống hiện thông báo lỗi | |

##### Tạo mới phòng ban.

|  |  |
| --- | --- |
| **Tác nhân:** Quản trị viên | **Loại UC:** business use-case |
| **Mô tả tổng quát:**  Usecase cho phép quản trị viên tạo mới thêm phòng ban | |
| **Kích hoạt:** Quản trị viên muốn tạo mới phòng ban mới | |
| **Mối quan hệ:**   * Người dùng thực hiện chức năng tạo mới phòng ban * Tạo mới phòng ban là chức năng mở rộng của use case quản lý phòng ban | |
| **Luồng xử lý chính:**   1. Quản trị viên đăng nhập vào hệ thống chọn “Quản Trị”. 2. Chọn “ Quản lý phòng ban”. 3. Chọn “ Tạo mới phòng ban” 4. Điền thông tin và nhấn tạo mới. | |
| **Luồng xử lý ngoại lệ:**  4b. Hệ thống xác thực thông tin đăng ký không thành công và hiển thị thông báo.  Usecase dừng lại  4b1. Quản trị viên sửa lại thông tin tài khoản cần tạo. Usecase tiếp tục Usecase 4 | |

##### Sửa phòng ban

|  |  |
| --- | --- |
| **Tác nhân:** Quản trị viên | **Loại UC:** business use-case |
| **Mô tả tổng quát:**  Usecase cho phép quản trị viên sửa thông tin phòng ban đang tồn tại | |
| **Kích hoạt:** Quản trị viên muốn sửa thông tin phòng ban đang tồn tại | |
| **Mối quan hệ:**   * Người dùng thực hiện chức năng sửa thông tin phòng ban * Sửa thông tin phòng ban là chức năng mở rộng của use case quản lý phòng ban | |
| **Luồng xử lý chính:**   1. Quản trị viên đăng nhập vào hệ thống chọn “Quản Trị”. 2. Chọn “ Quản lý phòng ban” 3. Chọn “ Xem và sửa thông tin phòng ban”. 4. Chọn biểu tượng “cây viết” nếu muốn sửa thông tin phòng ban đó. 5. Điền các thông tin cần chỉnh sửa 6. Nhấn “ Lưu” để cập nhật các thông tin đã chỉnh sửa | |
| **Luồng sửa lý thay thế:**  6b. Nhấn “ Huỷ” để trở lại trang xem danh sách phòng ban | |
| **Luồng xử lý ngoại lệ:**  6c. Hệ thống xác thực thông tin đăng ký không thành công và hiển thị thông báo.  Usecase dừng lại  6b1. Quản trị viên sửa lại thông tin tài khoản chỉnh sửa. Usecase tiếp tục Usecase 4 | |

##### Xoá phòng ban.

|  |  |
| --- | --- |
| **Tác nhân:** Quản trị viên | **Loại UC:** business use-case |
| **Mô tả tổng quát:**  Usecase cho phép quản trị viên xoá những phòng ban đang tồn tại | |
| **Kích hoạt:** Quản trị viên muốn xoá phòng ban đang tồn tại | |
| **Mối quan hệ:**   * Người dùng thực hiện chức năng xoá phòng ban * Xoá phòng ban là chức năng mở rộng của use case quản lý phòng ban | |
| **Luồng xử lý chính:**   1. Quản trị viên đăng nhập vào hệ thống chọn “Quản Trị”. 2. Chọn “ Quản lý phòng ban”. 3. Chọn “ Xem và sửa phòng ban” 4. Chọn biểu tượng “thùng rác” nếu muốn xoá ban đó. | |
| **Luồng xử lý ngoại lệ:** Khi lỗi server/mất mạng xảy ra người dùng không thể truy cập và chỉnh sửa nhật ký, hệ thống hiện thông báo lỗi | |

#### Quản lý bài đăng

##### Xem danh sách bài đăng, bản nháp cá nhân

|  |  |
| --- | --- |
| **Tác nhân:** sinh viên | **Loại UC:** business use-case |
| **Mô tả tổng quát:** Usecase cho phép sinh viên xem danh sách bài đăng theo từng loại của cá nhân mình. | |
| **Kích hoạt:** Sinh viên muốn xem danh sách bài đăng theo từng loại của cá nhân mình | |
| **Mối quan hệ:**   * Sinh viên thực hiện chức năng xem danh sách bài đăng, bản nháp cá nhân. * Xem bài đăng cá nhân là chức năng mở rộng của use case quản lý bài đăng | |
| **Luồng xử lý chính:**   1. Sinh viên đăng nhập vào hệ thống chọn “Quản lý bài đăng” 2. Chọn loại bài đăng muốn xem là “ Tổng quan”, “ Đã duyệt”, “ Chờ duyệt”, “ Bản nháp”, “ Spam”. | |
| **Luồng xử lý ngoại lệ:**  Khi lỗi server/mất mạng xảy ra người dùng không thể truy cập và chỉnh sửa nhật ký, hệ thống hiện thông báo lỗi | |

##### Xem chi tiết bài đăng cá nhân.

|  |  |
| --- | --- |
| **Tác nhân:** Sinh viên | **Loại UC:** business use-case |
| **Mô tả tổng quát:**  Usecase cho phép sinh viên xem chi tiết bài đăng theo từng loại của cá nhân mình. | |
| **Kích hoạt:** Sinh viên muốn xem chi tiết thông tin các bài đăng của mình | |
| **Mối quan hệ:**   * Sinh viên thực hiện chức năng xem chi tiết bài đăng cá nhân * Chức năng xem chi tiết bài đăng cá nhân là chức năng mở rộng của quản lý bài đăng | |
| **Luồng xử lý chính:**   1. Sinh viên đăng nhập vào hệ thống chọn “Quản lý bài đăng” 2. Chọn loại bài đăng muốn xem là “ Tổng quan”, “ Đã duyệt”, “ Chờ duyệt”, “ Bản nháp”, “ Spam”. 3. Chọn bài đăng muốn xem. | |
| **Luồng xử lý ngoại lệ:** Khi lỗi server/mất mạng xảy ra người dùng không thể truy cập và chỉnh sửa nhật ký, hệ thống hiện thông báo lỗi | |

##### Tạo bản nháp

|  |  |
| --- | --- |
| **Tác nhân:** Sinh viên | **Loại UC:** business use-case |
| **Mô tả tổng quát:**  Usecase cho phép sinh viên tạo bản nháp cho bài đăng mới | |
| **Kích hoạt:** Sinh viên muốn tạo bản nháp cho bài đăng mới | |
| **Mối quan hệ:**   * Sinh viên thực hiện chức năng tạo bản nháp * Chức năng tạo bản nháp là chức năng mở rộng của quản lý bài đăng | |
| **Luồng xử lý chính:**   1. Sinh viên đăng nhập vào hệ thống chọn “Quản lý bài đăng” 2. Chọn “Tạo bài đăng mới” 3. Chọn biểu tương dấu “+” để tạo mới. 4. Điền thông tin cần tạo bản nháp 5. Nhấn “ Lưu bản nháp” để lưu lại những thông tin cần lưu. | |
| **Luồng xử lý ngoại lệ:** Khi lỗi server/mất mạng xảy ra người dùng không thể truy cập và chỉnh sửa nhật ký, hệ thống hiện thông báo lỗi | |

##### Sửa bản nháp

|  |  |
| --- | --- |
| **Tác nhân:** Sinh viên | **Loại UC:** business use-case |
| **Mô tả tổng quát:**  Usecase cho phép sinh viên sửa những bản nháp đang tồn tại | |
| **Kích hoạt:** Sinh viên muốn sửa lại những bản nháp đang tồn tại | |
| **Mối quan hệ:**   * Sinh viên thực hiện chức năng sửa bản nháp * Chức năng sửa bản nháp là chức năng mở rộng của quản lý bài đăng | |
| **Luồng xử lý chính:**   1. Sinh viên đăng nhập vào hệ thống chọn “Quản lý bài đăng” 2. Chọn “ Tổng quan” 3. Danh sách các bài đăng được hiện ra 4. Chọn bản nháp muốn sửa 5. Sửa các thông tin muốn chỉnh sửa 6. Nhấn “ Lưu bản nháp” để lưu lại các thông tin đã sửa | |
| **Luồng xử lý thay thế:**  2b. Chọn “Tạo bài đăng mới”. Usecase tiếp tục | |
| **Luồng xử lý ngoại lệ:**  Khi lỗi server/mất mạng xảy ra người dùng không thể truy cập và chỉnh sửa nhật ký, hệ thống hiện thông báo lỗi | |

##### Xoá bản nháp

|  |  |
| --- | --- |
| **Tác nhân:** Sinh viên | **Loại UC:** business use-case |
| **Mô tả tổng quát:** Usecase cho phép sinh viên xoá những bản nháp đang tồn tại | |
| **Kích hoạt:** Sinh viên muốn xoá những bản nháp đang tồn tại | |
| **Mối quan hệ:**   * Sinh viên thực hiện chức năng xoá bản nháp * Chức năng xoá bản nháp là chức năng mở rộng của quản lý bài đăng | |
| **Luồng xử lý chính:**   1. Sinh viên đăng nhập vào hệ thống chọn “Quản lý bài đăng” 2. Chọn “Tổng quan” 3. Chọn biểu tượng “thùng rác” bên cạnh các bản nháp đã có trước đó nếu muốn xoá. | |
| **Luồng xử lý thay thế:**  2b. Chọn “Tạo bài đăng mới”. Usecase tiếp tục | |
| **Luồng xử lý ngoại lệ:**  Khi lỗi server/mất mạng xảy ra người dùng không thể truy cập và chỉnh sửa nhật ký, hệ thống hiện thông báo lỗi | |

##### Gửi bài đăng chờ kiểm duyệt

|  |  |
| --- | --- |
| **Tác nhân:** Sinh viên | **Loại UC:** business use-case |
| **Mô tả tổng quát**: Usecase cho phép sinh viên gửi những bản nháp hoặc bài đăng mới tạo để cho nhân viên phòng ban duyệt bài và public. | |
| **Kích hoạt:** Sinh viên muốn gửi những bản nháp hoặc bài đăng mới để cho nhân viên phòng ban duyệt bài và public | |
| **Mối quan hệ:**   * Sinh viên thực hiện chức năng gửi bài đăng chờ kiểm duyệt * Chức năng gửi bài đăng chờ kiểm duyệt là chức năng mở rộng của quản lý bài đăng | |
| **Luồng xử lý chính:**   1. Sinh viên đăng nhập vào hệ thống chọn “Quản lý bài đăng” 2. Chọn “Tạo mới đăng mới” 3. Chọn tạo mới hoặc chọn bản nháp đã có sẵn. 4. Nhấn nút “ Gửi” sau khi đã điền đầy đủ các thông tin | |
| **Luồng xử lý ngoại lệ:**  4b. Hệ thống xác thực thông tin import không thành công và hiển thị thông báo.  Use case dừng lại  4. Sinh viên sửa lại thông tin tài khoản cần tạo. Usecase tiếp tục Usecase 4. | |

5.1.4.4.7 Tìm kiếm bài đăng

|  |  |
| --- | --- |
| **Tác nhân:** Khách vãn lai, sinh viên, nhân viên phòng ban, ban giám hiệu. | **Loại UC:** business use-case |
| **Mô tả tổng quát:**  Usecase cho phép người dùng truy cập vào trang web có thể tìm kiếm các bài đăng đã public. | |
| **Kích hoạt:** Người dùng muốn tìm kiếm các bài biết đã public | |
| **Mối quan hệ:**   * Sinh viên thực hiện chức năng gửi bài đăng chờ kiểm duyệt * Chức năng gửi bài đăng chờ kiểm duyệt là chức năng mở rộng của quản lý bài đăng | |
| **Luồng xử lý chính:**   1. Truy cập vào trang web. 2. Gõ vào ô tìm kiếm nội dung cần tìm và nhấn enter | |
| **Luồng xử lý ngoại lệ:**  Khi lỗi server/mất mạng xảy ra người dùng không thể truy cập và chỉnh sửa nhật ký, hệ thống hiện thông báo lỗi | |

5.1.4.4.8 Xem danh sách bài đăng công khai

|  |  |
| --- | --- |
| **Tác nhân:** Khách vãn lai, sinh viên, nhân viên phòng ban, ban giám hiệu. | **Loại UC:** business use-case |
| **Mô tả tổng quát:**  Usecase cho phép mọi người có thể xem các bài đăng đã public khi truy cập vào trang web | |
| **Kích hoạt:** Người dùng muốn xem các bài đăng đã public | |
| **Mối quan hệ:**   * Người dùng thực hiện chức năng xem danh sách bài đăng công khai * Chức năng Xem bài đăng công khai là chức năng mở rộng của quản lý bài đăng | |
| **Luồng xử lý chính:**   1. Truy cập vào hệ thống để thấy các bài đăng đã public. | |
| **Luồng xử lý ngoại lệ:**  Khi lỗi server/mất mạng xảy ra người dùng không thể truy cập và chỉnh sửa nhật ký, hệ thống hiện thông báo lỗi | |

5.1.4.4.9 Xem chi tiết bài đăng công khai

|  |  |
| --- | --- |
| **Tác nhân:** Khách vãn lai, sinh viên, nhân viên phòng ban, ban giám hiệu. | **Loại UC:** business use-case |
| **Mô tả tổng quát:**  Usecase cho phép mọi người có thể xem chi tiết các bài đăng đã public khi truy cập vào trang web. | |
| **Kích hoạt:** Người dùng muốn xem chi tiết các bài đăng đã public | |
| **Mối quan hệ:**   * Người dùng thực hiện chức năng xem chi tiết bài đăng công khai * Chức năng Xem bài đăng công khai là chức năng mở rộng của quản lý bài đăng | |
| **Luồng xử lý chính:**   1. Truy cập vào hệ thống để thấy các bài đăng đã public. 2. Chọn bài đăng muốn xem chi tiết. | |
| **Luồng xử lý ngoại lệ:**  Khi lỗi server/mất mạng xảy ra người dùng không thể truy cập và chỉnh sửa nhật ký, hệ thống hiện thông báo lỗi | |

#### Quản lý chứng thực

##### Đăng nhập

|  |  |
| --- | --- |
| **Tác nhân:** Sinh viên, nhân viên phòng ban, ban giám hiệu | **Loại UC:** business use-case |
| **Mô tả tổng quát:**  Là người dùng, tôi muốn đăng nhập vào website để sử dụng chức năng website | |
| **Kích hoạt:** Người dùng muốn đăng nhập vào trang web | |
| **Mối quan hệ:**   * Người dùng thực hiện chức năng đăng nhập * Chức năng đăng nhập là chức năng mở rộng của quản lý chứng thực | |
| **Luồng xử lý chính:**   1. Người dùng truy cập vào trang web 2. Người dùng nhập thông tin tài khoản và chọn “ Đăng nhập”. 3. Hệ thống xác thực thông tin đăng nhập thành công và cho phép người dùng truy cập ứng dụng 4. Hệ thống ghi nhận hoạt động đăng nhập thành công vào Activity Log | |
| **Luồng xử lý ngoại lệ:**  3b. Hệ thống xác thực thông tin đăng nhập không thành công và hiển thị thông báo. Usecase dừng lại. | |

##### Đăng xuất

|  |  |
| --- | --- |
| **Tác nhân:** Sinh viên, nhân viên phòng ban, ban giám hiệu | **Loại UC:** business use-case |
| **Mô tả tổng quát:** Là người dùng, tôi muốn đăng xuất khỏi trang web | |
| **Kích hoạt:** Người dùng muốn đăng xuất khỏi trang web | |
| **Mối quan hệ:**   * Người dùng thực hiện chức năng Đăng xuất * Đăng xuất là chức năng mở rộng của quản lý chứng thực | |
| **Luồng xử lý chính:**  1. Người dùng truy cập và đăng nhập vào trang web thành công  2. Người dùng ấn vào icon tên của mình góc bên phải phía trên ở màn hình  3. Người dùng chọn lệnh Đăng xuất  4. Người dùng đăng xuất khỏi trang web | |
| **Luồng xử lý ngoại lệ:**  Khi lỗi server/mất mạng xảy ra người dùng không thể đăng xuất khỏi ứng dụng và hiện thị thông báo lỗi server | |

##### Đổi mật khẩu

|  |  |
| --- | --- |
| **Tác nhân:** Đối tượng sử dụng bao gồm các thành viên trong use case đăng nhập. | **Loại UC:** business use-case |
| **Mô tả tổng quát:** Là người dùng, tôi muốn đổi mật khẩu tài khoản của mình | |
| **Kích hoạt:** Người dùng muốn đổi mật khẩu của tài khoản | |
| **Mối quan hệ:**   * Người dùng thực hiện chức năng đổi mật khẩu * Đăng xuất là chức năng mở rộng của quản lý chứng thực | |
| **Luồng xử lý chính:**  1. Người dùng truy cập và đăng nhập vào trang web thành công  2. Người dùng ấn vào icon tên của mình góc bên phải phía trên ở màn hình  3. Người dùng chọn lệnh Đăng xuất  4. Người dùng đăng xuất khỏi trang web | |
| **Luồng xử lý ngoại lệ:**  Khi lỗi server/mất mạng xảy ra người dùng không thể đăng xuất khỏi ứng dụng và hiện thị thông báo lỗi server | |

#### Kiểm duyệt bài đăng

##### Xem bài đăng và chờ duyệt

|  |  |
| --- | --- |
| **Tác nhân:** Nhân viên phòng ban | **Loại UC:** business use-case |
| **Mô tả tổng quát:** Usecase này cho phép nhân viên phòng ban xem bài đăng được sinh viên gửi tới để chờ duyệt public. | |
| **Kích hoạt:** Nhân viên phòng ban muốn xem bài đăng được sinh viên gửi đến để duyệt public | |
| **Mối quan hệ:**   * Nhân viên phòng ban thực hiện chức năng xem bài đăng chờ duyệt * Xem bài đăng chờ duyệt là chức năng mở rộng của kiểm duyệt bài đăng | |
| **Luồng xử lý chính:**   1. Nhân viên phòng ban đăng nhâp vào hệ thống chọn “ Duyệt bài đăng”. 2. Chọn bài đăng đang chờ duyệt để xem nội dung. | |
| **Luồng xử lý ngoại lệ:**  Khi lỗi server/mất mạng xảy ra người dùng không thể đăng xuất khỏi ứng dụng và hiện thị thông báo lỗi server | |

##### Phản hồi và duyệt public

|  |  |
| --- | --- |
| **Tác nhân:** Nhân viên phòng ban | **Loại UC:** business use-case |
| **Mô tả tổng quát:** Usecase này cho phép nhân viên phòng ban duyệt public các bài đăng ở trạng thái chờ duyệt khi thấy nó hợp lệ. | |
| **Kích hoạt:** Nhân viên phòng ban muốn duyệt public các bài đăng ở trạng thái chờ duyệt khi thấy nó hợp lệ | |
| **Mối quan hệ:**   * Nhân viên phòng ban thực hiện chức năng phản hồi và duyệt public * Phản hồi và duyệt public là chức năng mở rộng của kiểm duyệt bài đăng | |
| **Luồng xử lý chính:**   1. Nhân viên phòng ban đăng nhâp vào hệ thống chọn “ Duyệt bài đăng”. 2. Chọn bài đăng đang chờ duyệt để xem nội dung. 3. Chọn nút “ Duyệt” nếu thấy bài đăng hợp lệ. | |
| **Luồng xử lý ngoại lệ:**  Khi lỗi server/mất mạng xảy ra người dùng không thể đăng xuất khỏi ứng dụng và hiện thị thông báo lỗi server | |

##### Từ chối

|  |  |
| --- | --- |
| **Tác nhân:** Nhân viên phòng ban | **Loại UC:** business use-case |
| **Mô tả tổng quát:** Usecase này cho phép nhân viên phòng ban từ chối giải quyết bài đăng đó nếu thấy nó nằm ngoài thẩm quyền cũng như khả năng của nhân viên kiểm duyệt, và hệ thống sẽ gửi mail thông báo cho sinh viên đã gửi bài đăng để hẹn gặp mặt trực tiếp giải quyết. | |
| **Kích hoạt:** Nhân viên phòng ban muốn từ chối giải quyết bài đăng đang ở trạng thái chờ duyệt | |
| **Mối quan hệ:**   * Nhân viên phòng ban thực hiện chức năng từ chối * Từ chối là chức năng mở rộng của kiểm duyệt bài đăng | |
| **Luồng xử lý chính:**   1. Nhân viên phòng ban đăng nhâp vào hệ thống chọn “ Duyệt bài đăng”. 2. Chọn các bài đăng đang chờ duyệt để xem nội dung. 3. Chọn nút “ Từ chối” nếu thấy bài đăng nằm ngoài khả năng giải quyết. 4. Hệ thống tự động gửi mail thông báo cho sinh viên tình trạng của bài viết. | |
| **Luồng xử lý ngoại lệ:**  Khi lỗi server/mất mạng xảy ra người dùng không thể đăng xuất khỏi ứng dụng và hiện thị thông báo lỗi server | |

##### Đánh spam và xoá

|  |  |
| --- | --- |
| **Tác nhân:** Nhân viên phòng ban | **Loại UC:** business use-case |
| **Mô tả tổng quát:** Usecase này cho phép nhân viên phòng ban đánh spam bài đăng đó nếu thấy nó sử dụng những từ ngữ không phù hợp hoặc đã có những bài đăng tương tự, sinh viên sẽ nhận được email thông báo sau về tình trạng bài đăng của mình**.** | |
| **Kích hoạt:** Nhân viên phòng ban muốn đánh spam và xoá bài đăng đang ở trạng thái chờ duyệt | |
| **Mối quan hệ:**   * Nhân viên phòng ban thực hiện chức năng đánh spam và xoá * Đánh spam và xoá là chức năng mở rộng của kiểm duyệt bài đăng | |
| **Luồng xử lý chính:**   1. Nhân viên phòng ban đăng nhâp vào hệ thống chọn “Duyệt bài đăng”. 2. Chọn bài đăng đang chờ duyệt để xem nội dung. 3. Chọn nút “Đánh spam và xoá” nếu thấy bài đăng đó có những từ ngữ không phù hợp hoặc đã có những bài đăng tương tự. 4. Hệ thống tự động gửi mail thông báo cho sinh viên tình trạng của bài viết. | |
| **Luồng xử lý ngoại lệ:**  Khi lỗi server/mất mạng xảy ra người dùng không thể đăng xuất khỏi ứng dụng và hiện thị thông báo lỗi server. | |

### Yêu cầu phi chức năng

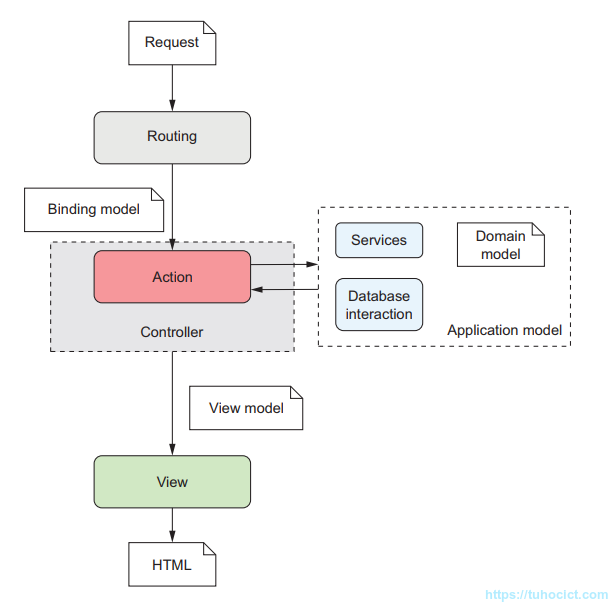
* Màu sắc của chủ đạo của website phải được dựa trên màu sắc hiện có của website Due.udn.vn của trường Đại học Kinh tế- Đại học Đà Nẵng, để thể hiện sự đồng bộ và thống nhất với website Due.udn.vn
* Hiệu năng của hệ thống khá tốt với thời gian phản hồi giữa các thao tác phải có phản hồi nhanh, không để người dùng đợi quá lâu.
* Bảo mật: Hệ thống phải được xây dựng dựa trên ASP.NET Core và bảo mật bằng Identity framework rất phổ biến hiện nay và có tính bảo mật khá tốt.
* Khả năng phục hồi: Dữ liệu của database phải được backup đều đặn, tránh mất dữ liệu không thể phục hồi.
* Khả năng sử dụng, tương tác: Hệ thống phải đơn giản hóa, dễ dàng sử dụng đối với người dùng.

## Thiết kế hệ thống

### Thiết kế kiến trúc hệ thống

Dự án sử dụng kiến trúc MVC của ASP.NET để xây dựng, gồm 3 phần chính:

* Views: là phần sẽ hiển thị bên phía client để thao tác, render các dữ liệu từ Models (database) theo yêu cầu của Controllers và truyền nó cho Client.
* Models: sẽ chứa các class về đối tượng trong hệ thống, ánh xạ từ database bằng Entity Framework, sau đó tương tác với database theo yêu cầu của Controllers được lấy từ phía Client (Views) sau đó truyền dữ liệu đã xử lý trả về cho Views để render ra màn hình cho Client.
* Controllers: Là phần nằm giữa và tương tác trực tiếp với Views và Models tương tự như Client và Database. Là nơi nhận request của Client từ Views sau đó thao tác dựa trên CRUB để tương tác với Models, cuối cùng gửi lại dữ liệu đã thao tác cho Views để truyền đến cho Client.



Hình 16 Mô hình MVC

Mô hình MVC sẽ hoạt động như sau:

1. Clients gọi yêu cầu xử lí từ Views.
2. Controllers nhận yêu cầu và đưa lệnh xử lí yêu cầu đó. Các lệnh thực thi với phần Views thì cập nhật hoặc phục vụ yêu cầu trang web, với Models thì để biểu đạt logic (Nếu lệnh yêu cầu có logic).
3. Models thực thi phần logic và lấy cơ sở dữ liệu rồi gửi trả lại phản hồi dựa trên hướng dẫn của Controllers.
4. Controllers truyền dữ liệu cho Views để cập nhận giao diện hiển thị cho người dùng.

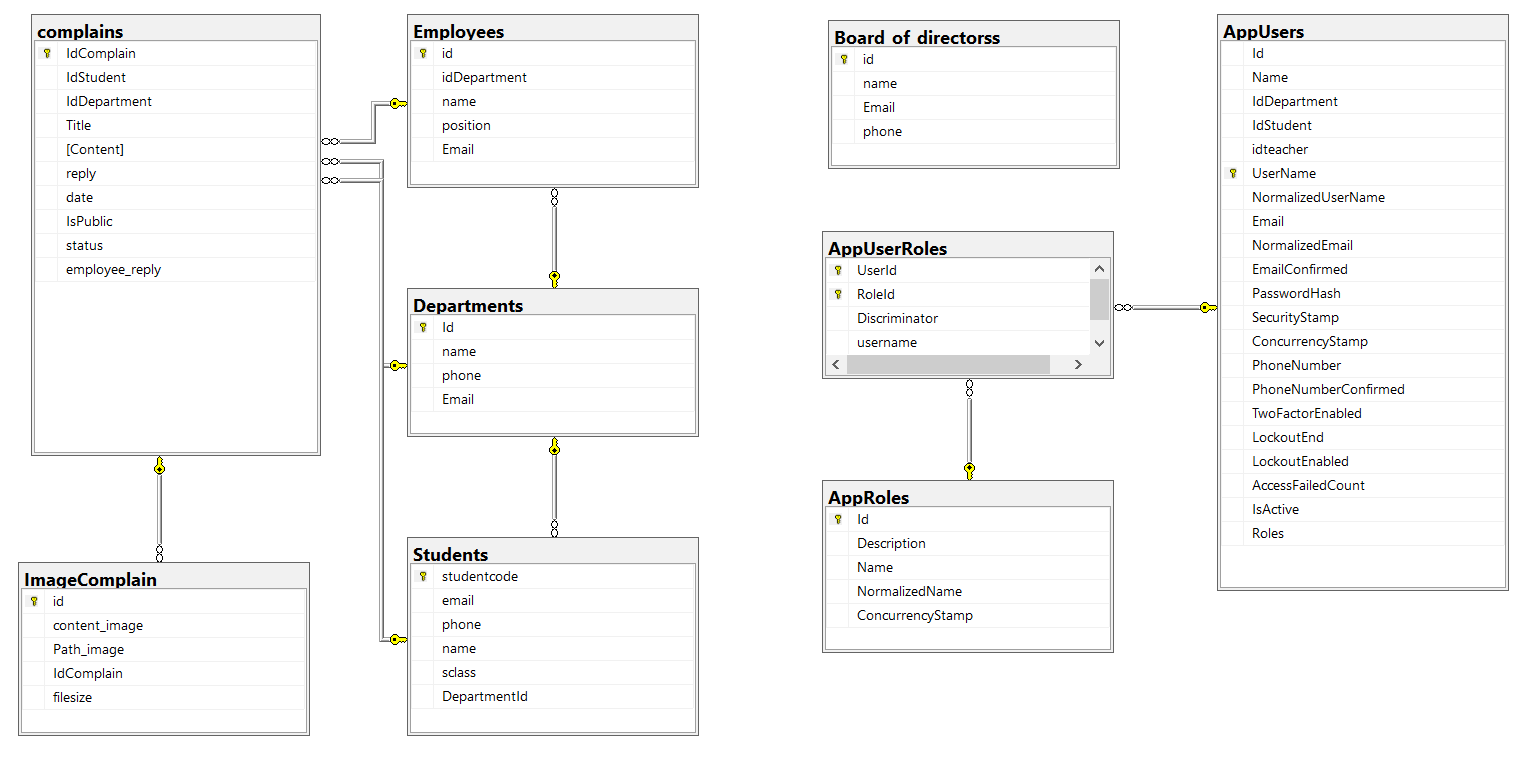
Database dự án dùng MS SQL 2018 sẽ ánh xạ 2 chiều bằng Entity Framework, có thể thao tác dữ liệu từ cả Server hoặc Client.

### **Thiết kế cơ sở dữ liệu**

Hệ thống sử dụng hệ quản trị cơ sở dữ liệu MS SQL 2018 với mô hình code first để thao tác và lưu trữ dữ liệu, được ánh xạ 2 chiều bằng Entity Framework. Kế thừa một số bảng từ IdentityDbContext, tuy nhiên đã sửa đổi sao cho phù hợp với nhu cầu của hệ thống.

Sử dụng mô hình code first sẽ giúp chúng ta thực hiện tất cả các công đoạn từ tạo cơ sở dữ liệu, tạo bảng, thêm bản ghi mẫu, … Và trong quá trình xây dựng hệ thống, nếu có bất cứ thay đổi nào phát sinh về cấu trúc các bảng, kiểu dữ liệu, các cột, … mô hình EF Code First cũng giúp chúng ta dễ dàng tuỳ ý thay đổi cấu trúc cơ sở dữ liệu để phù hợp với nhu cầu một cách nhanh chóng mà không cần dùng MS SQL cũng như có thể giữ lại các dữ liệu cũ.

Về tổng quan cơ sở dữ liệu như sau:



Hình 17. Tổng quan cơ sở dữ liệu

**5.3.3.1. Giải thích từng bảng của cơ sở dữ liệu**

|  |  |  |
| --- | --- | --- |
| STT | Tên bảng dữ liệu | Diễn giải |
| 1 | Board\_of\_Directorss | Lưu trữ dữ liệu thông tin ban giám hiệu |
| 2 | Employees | Lưu dữ liệu thông tin nhân viên của nhà trường |
| 3 | Departments | Lưu dữ liệu của các phòng ban |
| 4 | Students | Lưu dữ liệu thông tin sinh viên |
| 5 | Complains | Lưu dữ liệu các ý kiến, thắc mắc của sinh viên |
| 6 | ImageComplain | Lưu dữ liệu thông tin hình ảnh phản ánh |
| 7 | AppUsers | Lưu dữ liệu tài khoản người dùng |
| 8 | AppRoles | Lưu dữ liệu về vai trò |
| 9 | AppUserRoles | Lưu dữ liệu về phân quyền cho người dùng |

**5.3.3.2 Thiết kế chi tiết các bảng**

* **Bảng Board\_of\_Directorss**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** | **Ràng buộc** | **Chú thích** |
| 1 | id | int | Số nguyên | Khóa chính | Mã ban giám hiệu |
| 2 | name | Nvarchar(225) | Chuỗi từ 1 đến 225 kí tự |  | Tên của ban giám hiệu |
| 3 | email | Varchar(225) | Chuỗi từ 1 đến 225 kí tự |  | Email của ban giám hiệu |
| 4 | phone | Varchar(11) | Chuỗi từ 1 đến 11 kí tự |  | Số điện thoại liên lạc của ban giám hiệu |

* **Bảng Employees:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** | **Ràng buộc** | **Chú thích** |
| 1 | Id | Int | Số nguyên | Khóa chính | Mã số của nhân viên |
| 2 | IdDepartment | Int | Số nguyên | Khóa phụ | Mã các đơn vị trong trường |
| 3 | Name | Nvarchar(225) | Chuỗi từ 1 đến 255 kí tự. |  | Tên nhân viên |
| 4 | Position | Nvarchar(225) | Chuỗi từ 1 đến 255 kí tự. |  | Chức vụ của nhân viên |
| 5 | Email | Varchar(225) | Chuỗi từ 1 đến 128 kí tự, không trùng nhau, không để trống |  | Tên đăng nhập của nhân viên |

* **Bảng Departments:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** | **Ràng buộc** | **Chú thích** |
| 1 | Id | Int | Số nguyên | Khóa chính | Mã các đơn vị trong trường |
| 2 | Name | Nvarchar(225) | Chuỗi từ 1 đến 225 kí tự. |  | Tên đơn vị |
| 3 | Email | Varchar(max) | Số nguyên |  | Tổng số nhân viên |
| 3 | phone | Char(11) | Chuỗi từ 1 đến 11 kí tự |  | Số điện thoại của đơn vị |

* **Bảng Students:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** | **Ràng buộc** | **Chú thích** |
| 1 | Studentcode | Varchar(225) | Số nguyên | Khóa chính | Mã số sinh viên |
| 2 | Email | Varchar(256) | Chuỗi từ 1 đến 255 kí tự |  | Email của sinh viên |
| 3 | phone | Varchar(11) | Chuỗi từ 1 đến 11 kí tự |  | Số điện thoại của sinh viên |
| 4 | Name | Nvarchar(Max) | Chuỗi kí tự |  | Tên sinh viên |
| 5 | Class | Varchar(50) | Chuỗi 1 đến 50 kí tự |  | Lớp của sinh viên |
| 6 | DepartmentId | Int | Số nguyên | Khoá phụ | Đơn vị sinh viên trực thuộc |

* **Bảng Complains:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** | **Ràng buộc** | **Chú thích** |
| 1 | IdStudent | Varchar(225) | Số nguyên | Khóa phụ | Mã số viên viên |
| 2 | IdComplains | Int | Số nguyên | Khóa chính | Id bài viết của sinh viên |
| 3 | IdDepartment | Int | Số nguyên | Khóa phụ | Mã các đơn vị trong trường |
| 4 | Title | Nvarchar(225) | Chuỗi 1 đến 255 kí tự |  | Tiêu đề của bài viết |
| 5 | Content | Nvarchar(MAX) | Chuỗi |  | Nội dung bài viết |
| 5 | Reply | Nvarchar(Max) | Chuỗi |  | Kiểu bài viết |
| 6 | Date | Datetime | Thời gian |  | Thời gian viết bài |
| 7 | IsPublic | Bit | 1 hoặc 0 |  | Bài viết công khai / riêng tư |
| 8 | Status | Nvarchar(100) | Chuỗi kí tự từ 1 dến 100 |  | Tình trạng của bài viết |
| 9 | Employee\_reply | int | Số nguyên | Khoá phụ | Nhân viên phản hồi |

* **Bảng ImageComplain:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** | **Ràng buộc** | **Chú thích** |
| 1 | id | int | Số nguyên | Khóa chính | Mã số hình ảnh |
| 2 | Content\_Image | nvarchar(max) | Chuỗi |  | Nội dung hình ảnh |
| 3 | Path\_image | nvarchar(max) | Chuỗi |  | Đường dẫn hình ảnh |
| 4 | IdComplain | int | Số nguyên | Khoá phụ | Mã số phản ánh |
| 5 | filesize | int | Số nguyên |  | Kích thước hình ảnh |

* **Bảng AppUsers:** Bảng AppUsers kế thừa lại bảng IdentityUsers nên có sẵn rất nhiều cột, tuy nhiên chúng ta chỉ sử dụng một số cột chính sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** | **Ràng buộc** | **Chú thích** |
| 1 | Username | Nvarchar(450) | Chuỗi kí tự từ 1 đến 450 | Khóa chính | Tên đăng nhập người dùng |
| 2 | id | Nvarchar(max) | Chuỗi |  | Mã số tài khoản |
| 3 | Name | nvarchar(200) | Chuỗi kí tự từ 1 đến 200 |  | Tên người sỡ hữu tài khoản |
| 4 | IdDepartment | int | Số nguyên |  | Mã số phòng ban |
| 5 | IdStudent | Nvarchar(225) | Chuỗi kí tự từ 1 đến 225 |  | Mã số sinh viên |
| 6 | Idteacher | int | Số nguyên |  | Mã số giảng viên |
| 7 | Email | Nvarchar(max) | Chuỗi |  | Địa chỉ Email người dùng |
| 8 | IsActive | bit | 1 hoặc 0 |  | Kích hoạt/ khoá tài khoản người dùng |
| … |  |  |  |  |  |

* **Bảng AppRoles:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** | **Ràng buộc** | **Chú thích** |
| 1 | id | Nvarchar(450) | Chuỗi kí tự từ 1 đến 450 | Khóa chính | Mã số vai trò |
| 2 | Description | nvarchar(200) | Chuỗi kí tự từ 1 đến 200 |  | Mô tả vai trò |
| 3 | Name | nvarchar(max) | Chuỗi |  | Tên vai trò |
| 4 | NormalizedName | Nvarchar(max) | Chuỗi |  | Tên thông thường của vai trò |
| 5 | ConcurrencyStamp | Nvarchar(max) | Chuỗi |  | Một giá trị ngẫu nhiên sinh ra mỗi khi người dùng tác động đến bảng |

* **Bảng AppUserRoles:**

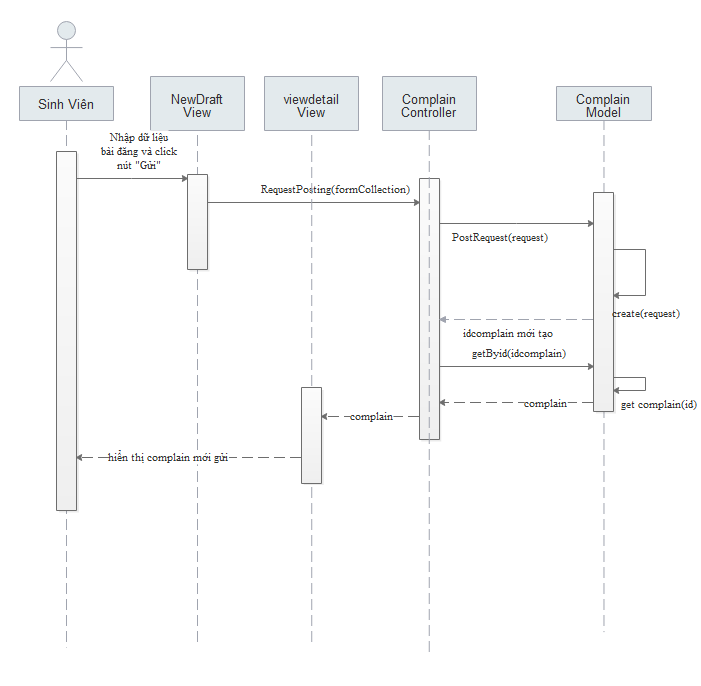
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Thuộc tính** | **Kiểu dữ liệu** | **Diễn giải** | **Ràng buộc** | **Chú thích** |
| 1 | Roleid | Nvarchar(450) | Chuỗi kí tự từ 1 đến 450 | Khóa chính | Mã số vai trò |
| 2 | Userid | nvarchar(450) | Chuỗi kí tự từ 1 đến 450 | Khoá chính | Mã số người dùng |
| 3 | Username | Nvarchar(450) | Chuỗi kí tự từ 1 đến 450 | Khoá phụ | Tên đăng nhập người dùng |

### Thiết kế chức năng

Hệ thống gồm 30 chức năng cho 5 actors, sau đây là một số chức năng chính của actor Sinh Viên, Ban Giám Hiệu và Nhân viên phòng ban.

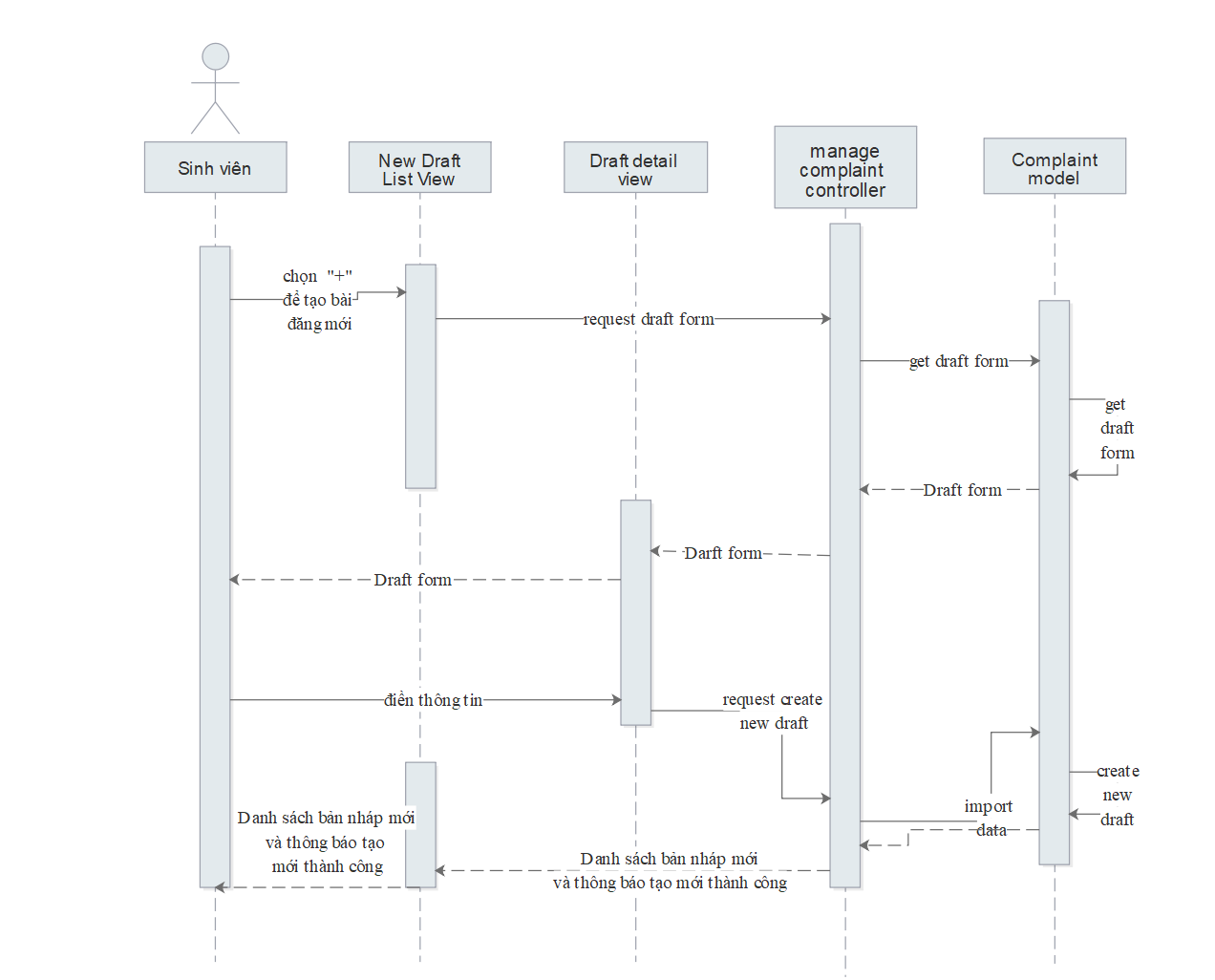
#### Sinh viên

##### Gửi bài đăng chờ duyệt



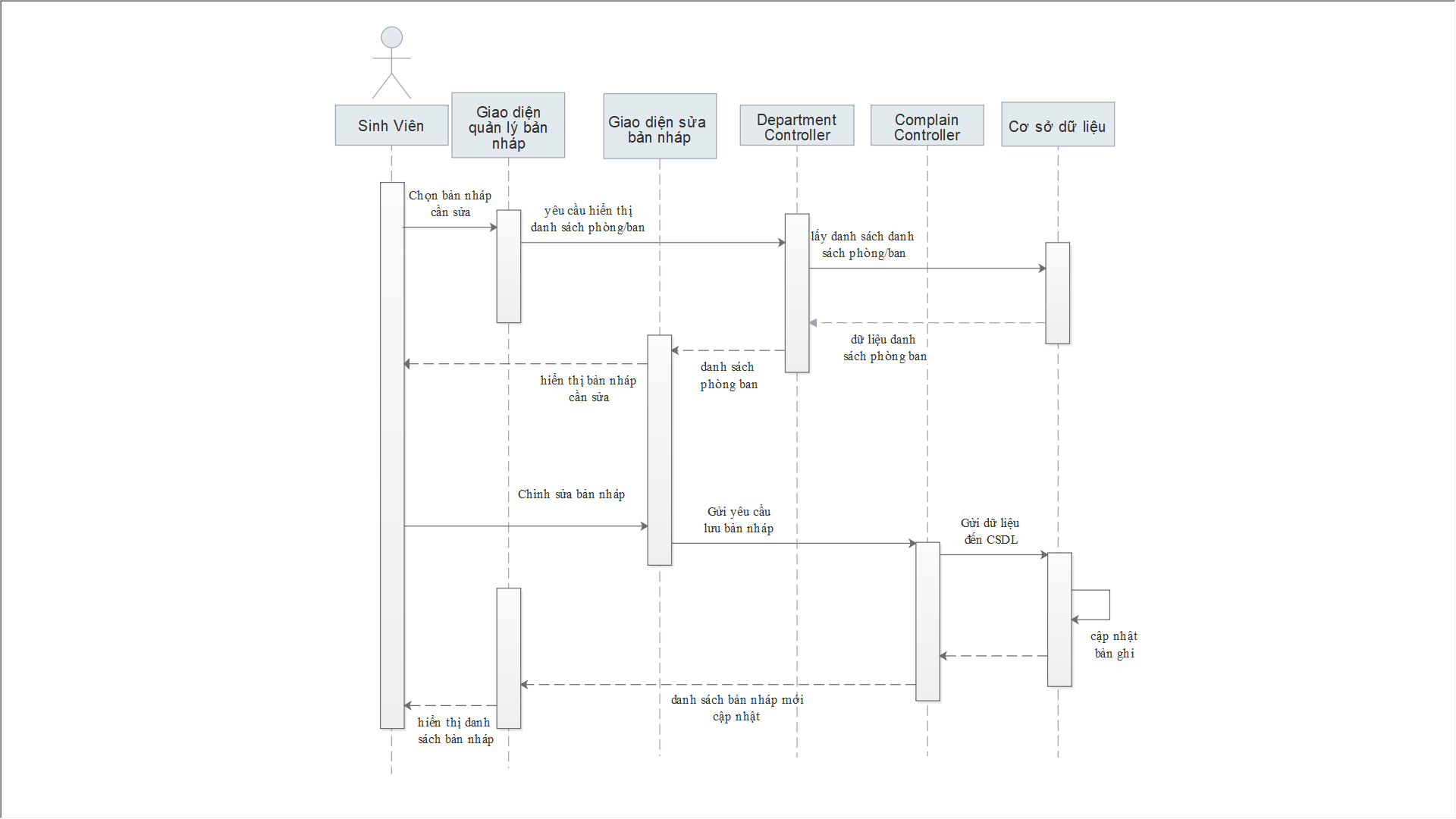
Hình 18 Chức năng gửi bài đăng chờ kiểm duyệt

##### Tạo bản nháp



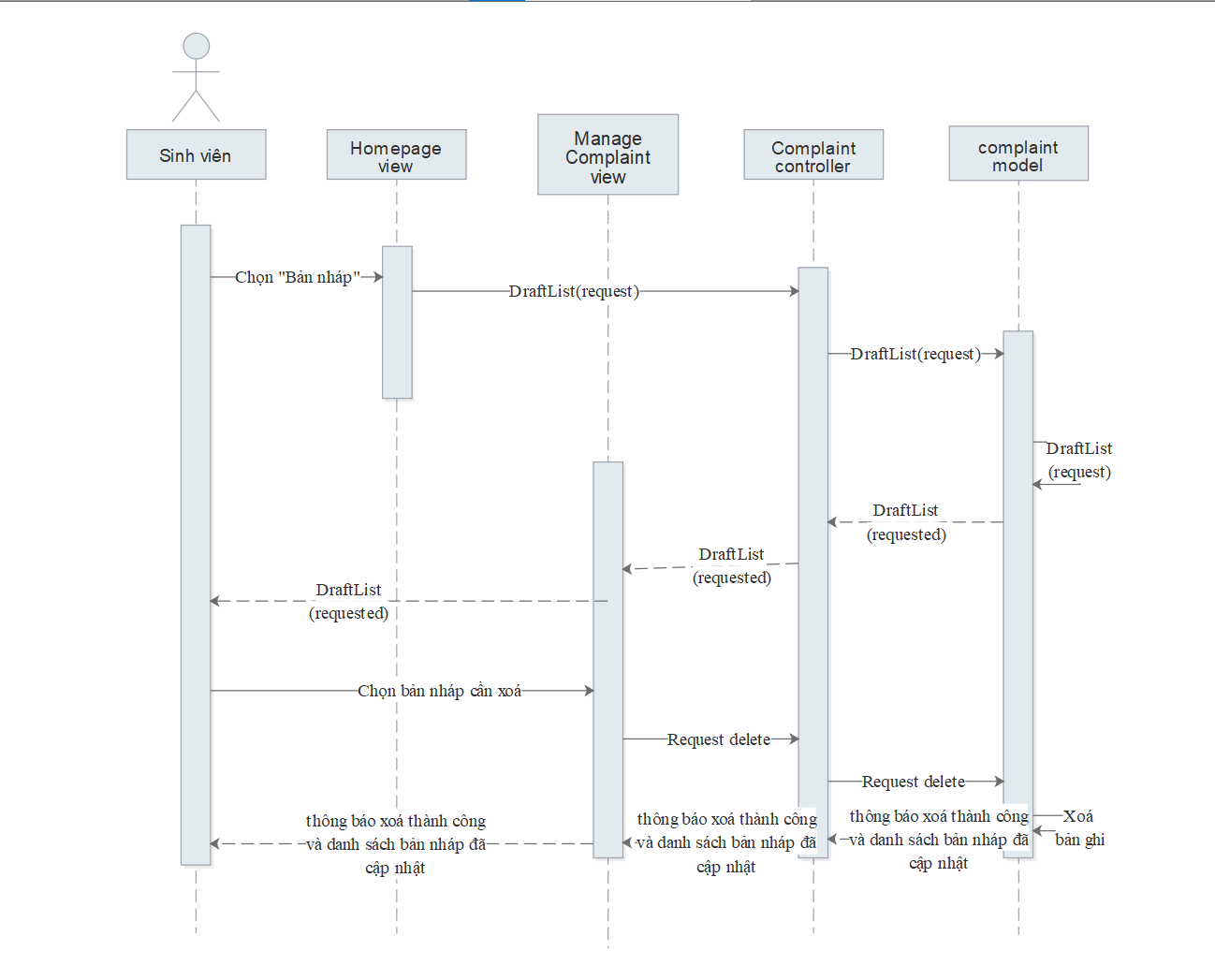
Hình 19 Chức năng tạo bản nháp

##### Sửa bản nháp



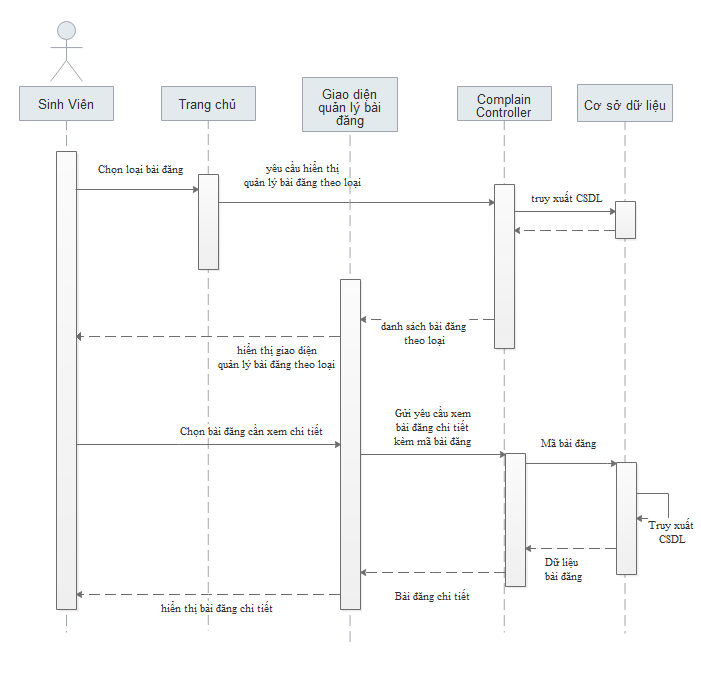
Hình 20 Chức năng sửa bản nháp

##### Xoá bản nháp



Hình 21 Chức năng xoá bản nháp

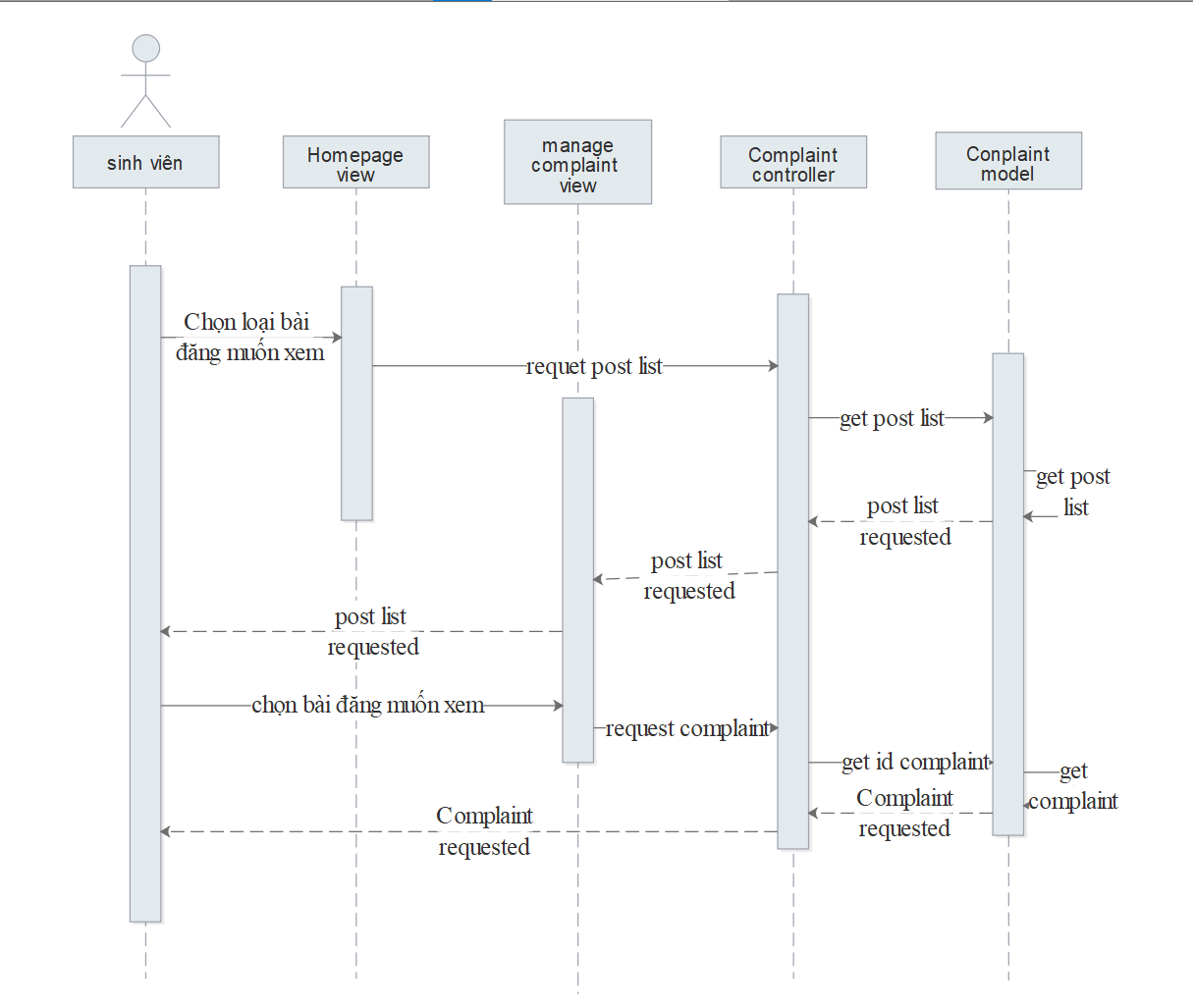
##### Xem danh sách bài đăng và xem bài đăng chi tiết



Hình 22 Chức năng xem danh sách bài đăng và xem bài đăng chi tiết

#### Nhân viên phòng ban

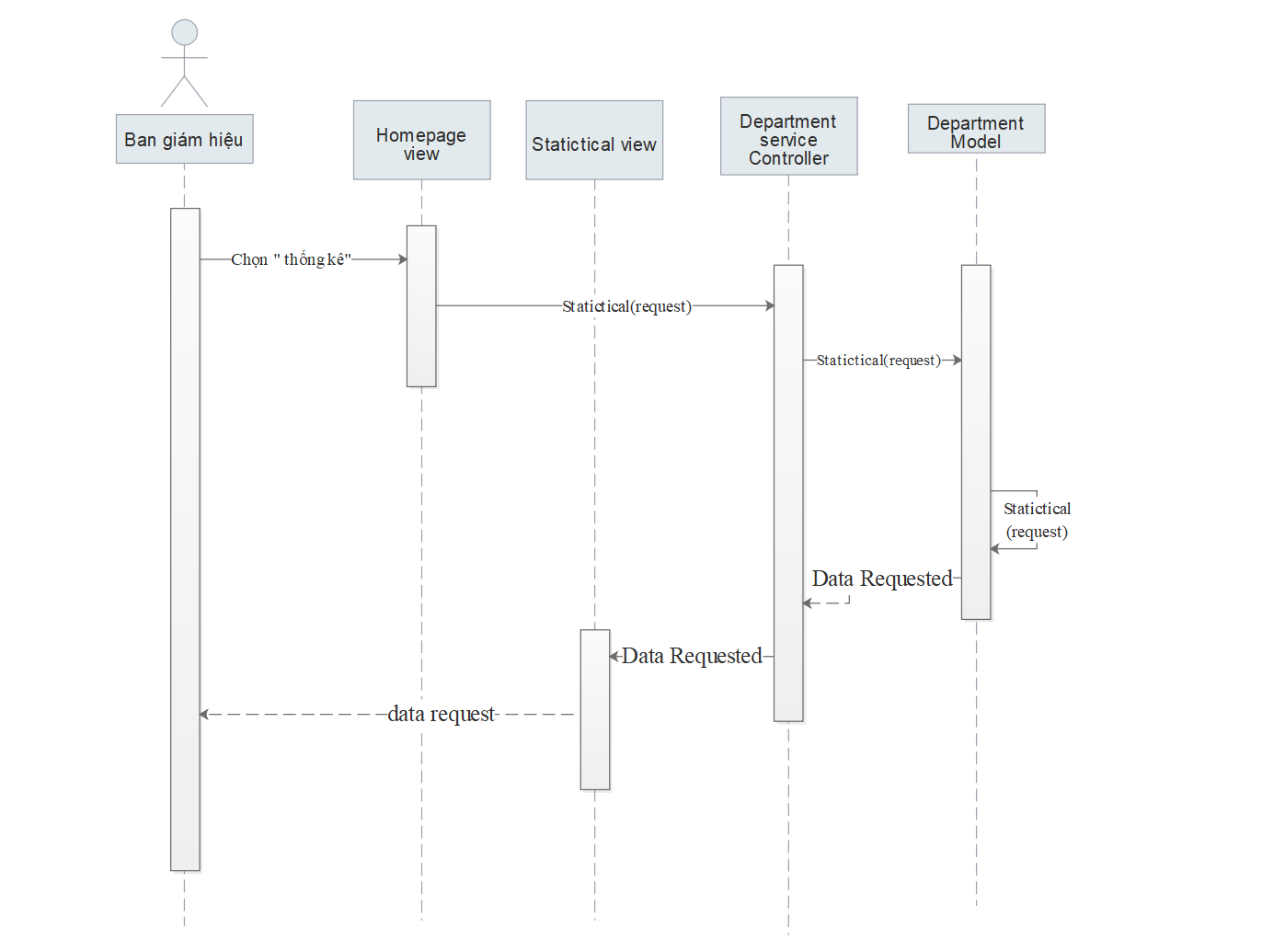
##### Xem danh sách bài đăng chờ duyệt, phản hồi và duyệt publish



Hình 23 Xem danh sách bài đăng chờ duyệt, phản hồi và duyệt publish

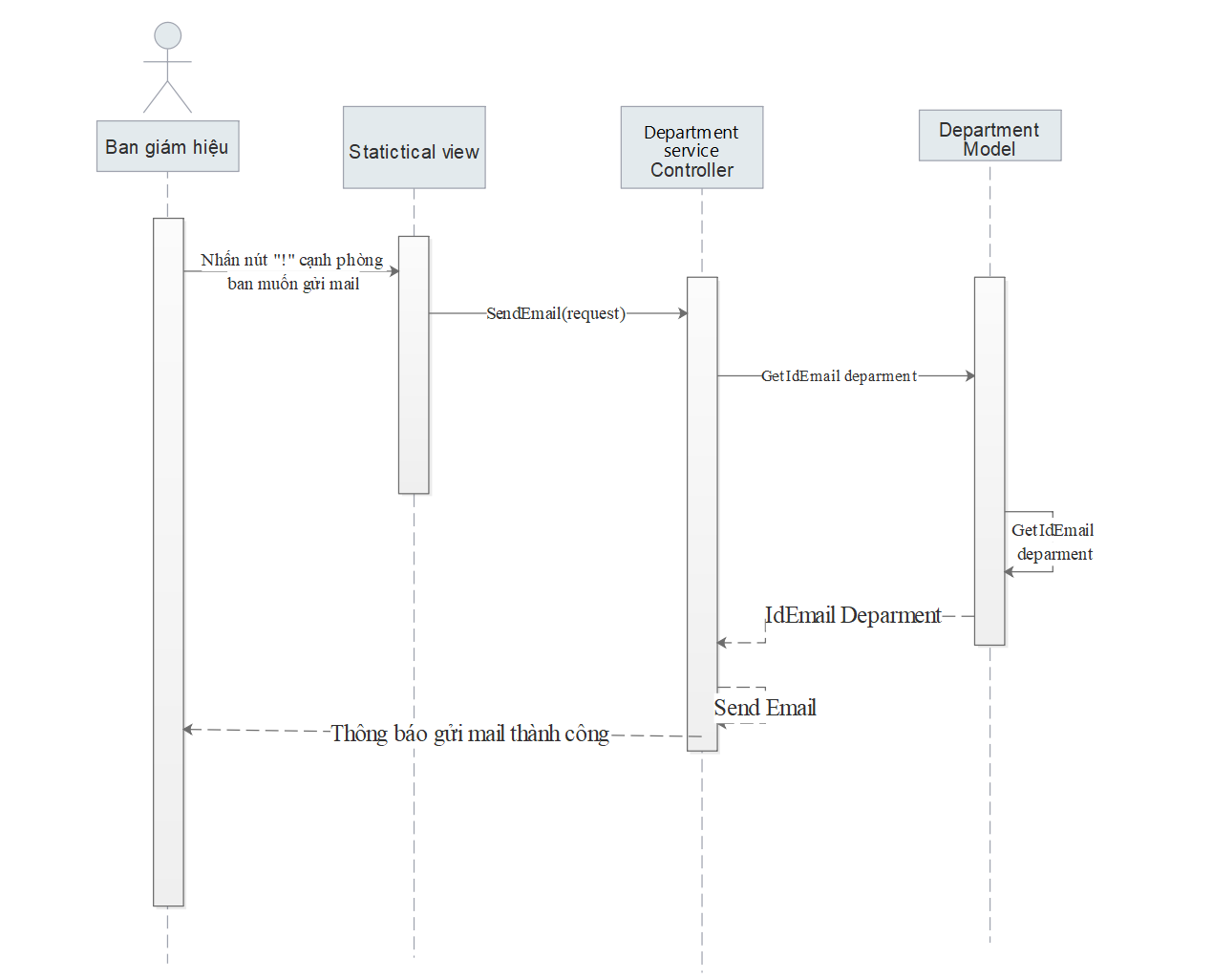
#### Ban giám hiệu

##### Xem thống kê bài đăng tổng quát



Hình 24 Chức năng xem thống kê bài đăng tổng quát

##### Gửi Mail cảnh báo cho phòng ban tiếp nhận phản ánh



Hình 25 Chức năng gửi mail cảnh báo

# PHÁT TRIỂN HỆ THỐNG

## Môi trường phát triển

### Môi trường phát triển của Front-end

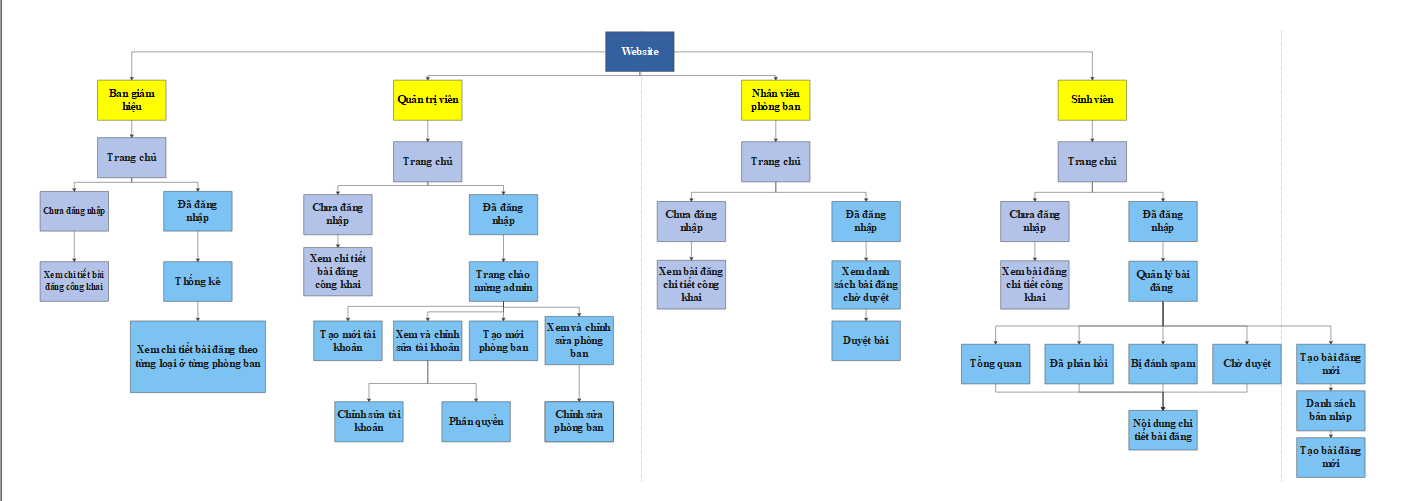
* Công cụ hỗ trợ: Visual studio code, Visual studio 2019
* Phần Front-end của hệ thống được phát triển dựa trên những kiến trúc cơ bản nhất của website đó là HTML, CSS, Javascript cũng với sợ hỗ trợ của Framework Bootstrap để có thể giúp thiết kế web chuẩn responsive một cách dễ dàng hơn.
* Vì phần nổi của website được cấu thành từ những thành phần cơ bản nhất nên giúp trang web có một tốc độ cũng như hiệu năng luôn ổn định.
* Dưới sự hỗ trợ của 1 trong những framework phổ biến nhất hiện nay là Boostrap để giúp giao diện trang web có thể thích hợp với mọi thiết bị hiện có trên thị trường.

### Môi trường phát triển của Back-end

* Hệ thống sử dụng ngôn ngữ nền tảng là C# nên sẽ sử dụng công cụ hỗ trợ Visual Studio để xây dựng ứng dụng. Sử dụng thêm một số Framework phổ biến để hỗ trợ như là: DotNet Core 3.1, Entity Framework, Swagger UI, …
* Sử dụng DotNet Core làm môi trường phát triển sẽ giúp dự án nhất quán trong việc lập trình hướng đối tượng có chức năng giống nhau cho dù code được lưu trữ và thực hiện cục bộ kể cả từ xa hay là trên server. Ngoài ra, còn loại bỏ phần lớn các xung đột phiên bản và triển khai phần mềm.
* Sử dụng Swagger UI sẽ giúp hệ thống build cho chúng ta một giao diện mô tả các RestAPI, giúp chúng ta có thể thực tương tác CRUD với các API mà không cần phải có Front-end.

## Thiết kế giao diện

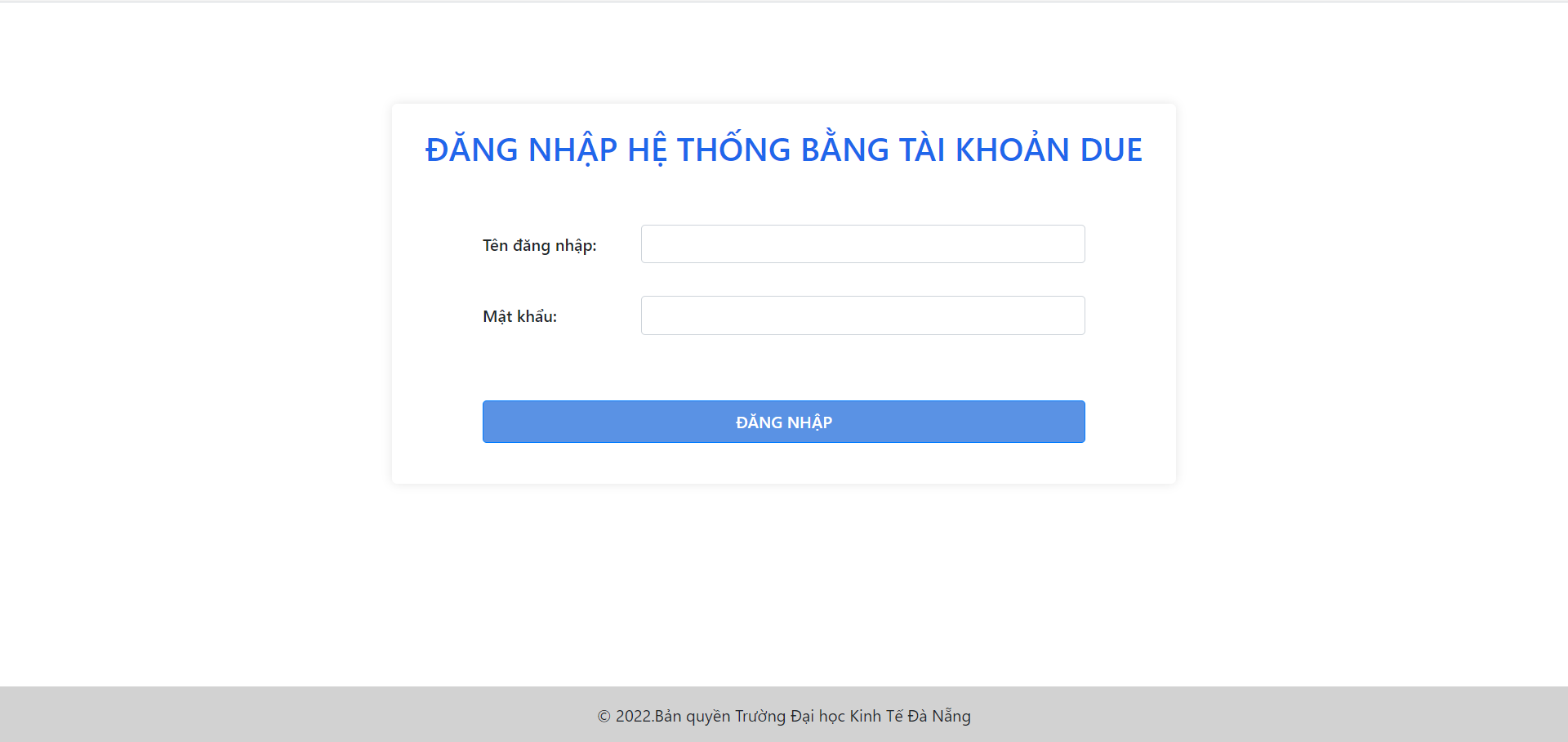
### Sơ đồ màn hình



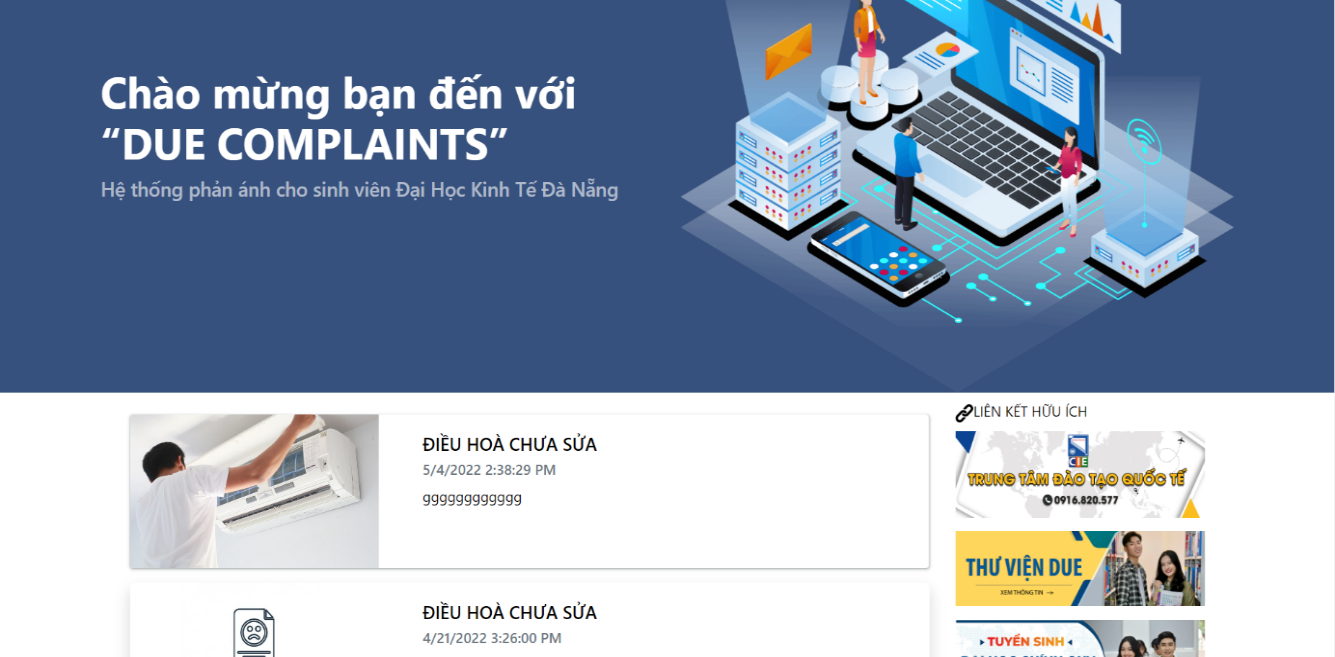
Hình 26 Sơ đồ màn hình

### Một số giao diện của hệ thống

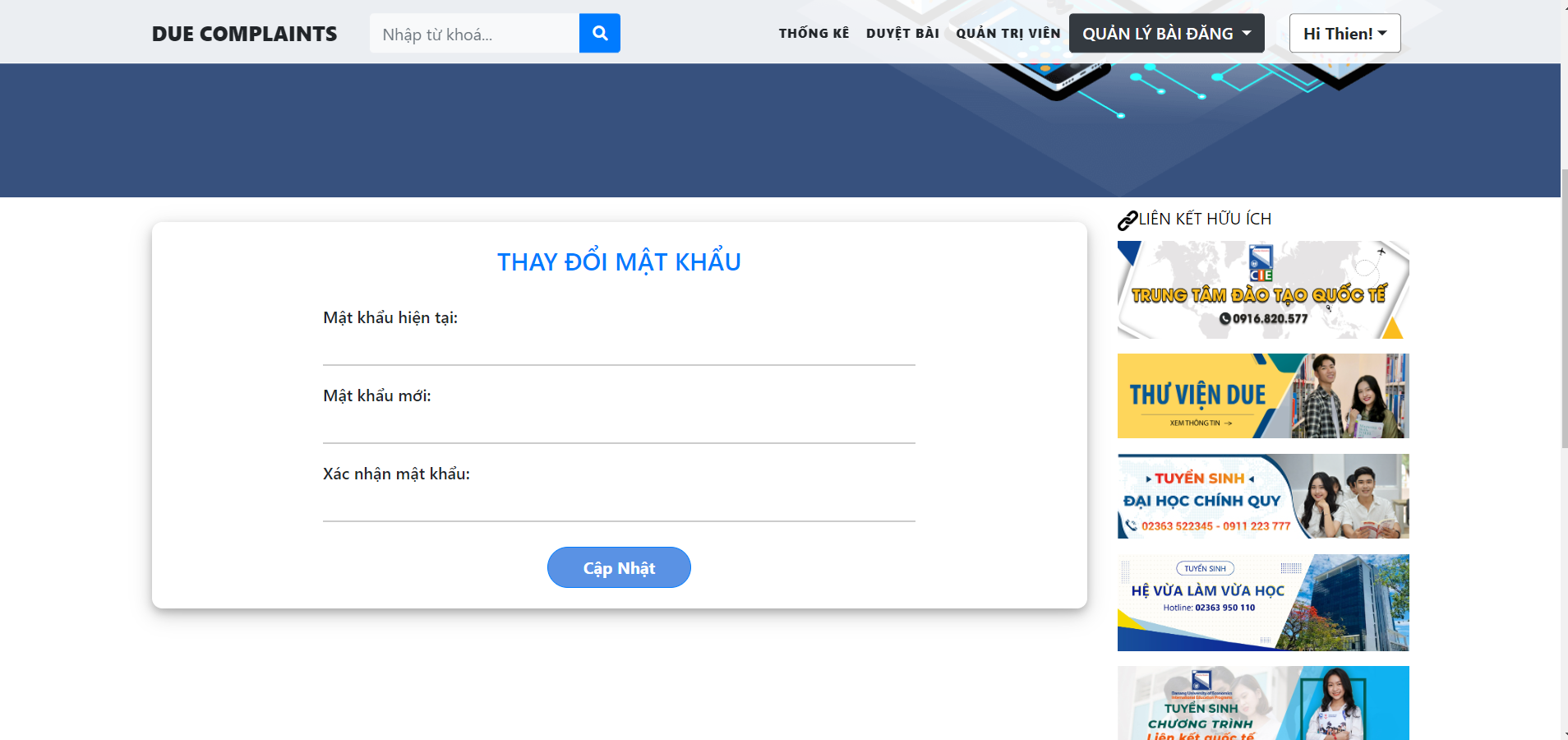
#### Các giao diện chính của khách và sinh viên



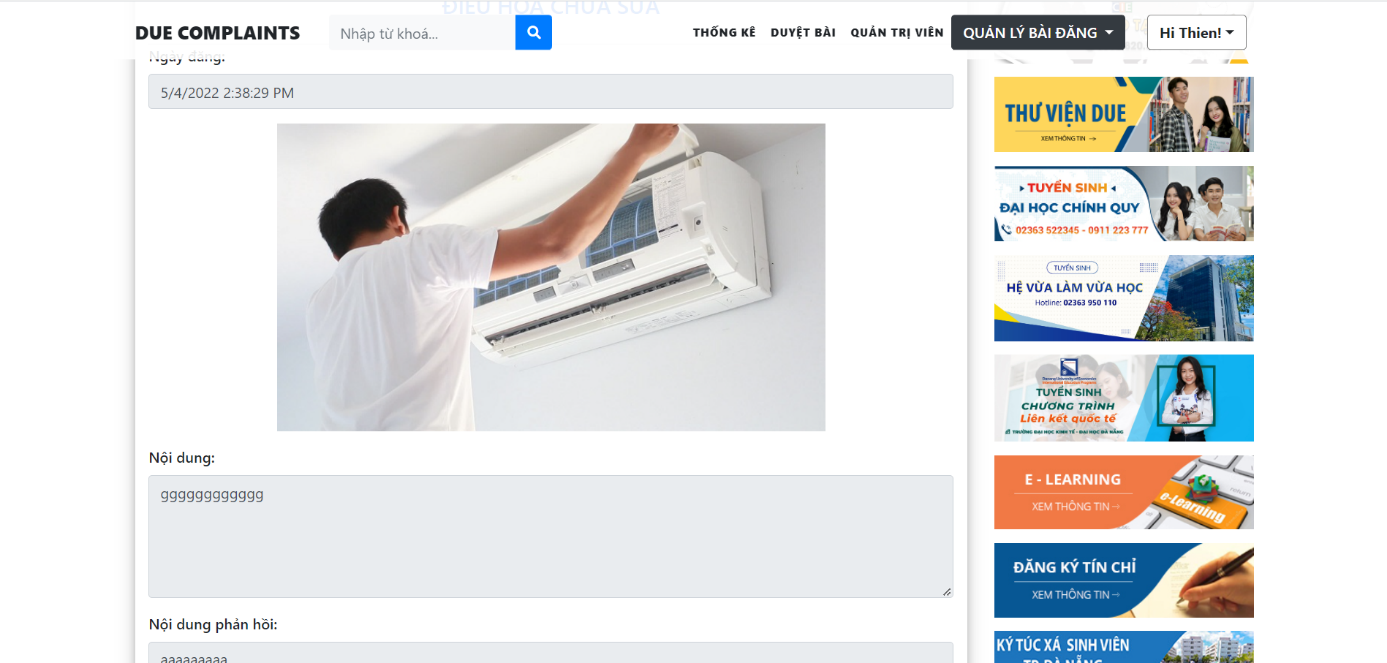
Hình 27. Giao diện trang đăng nhập



Hình 28. Giao diện trang chủ



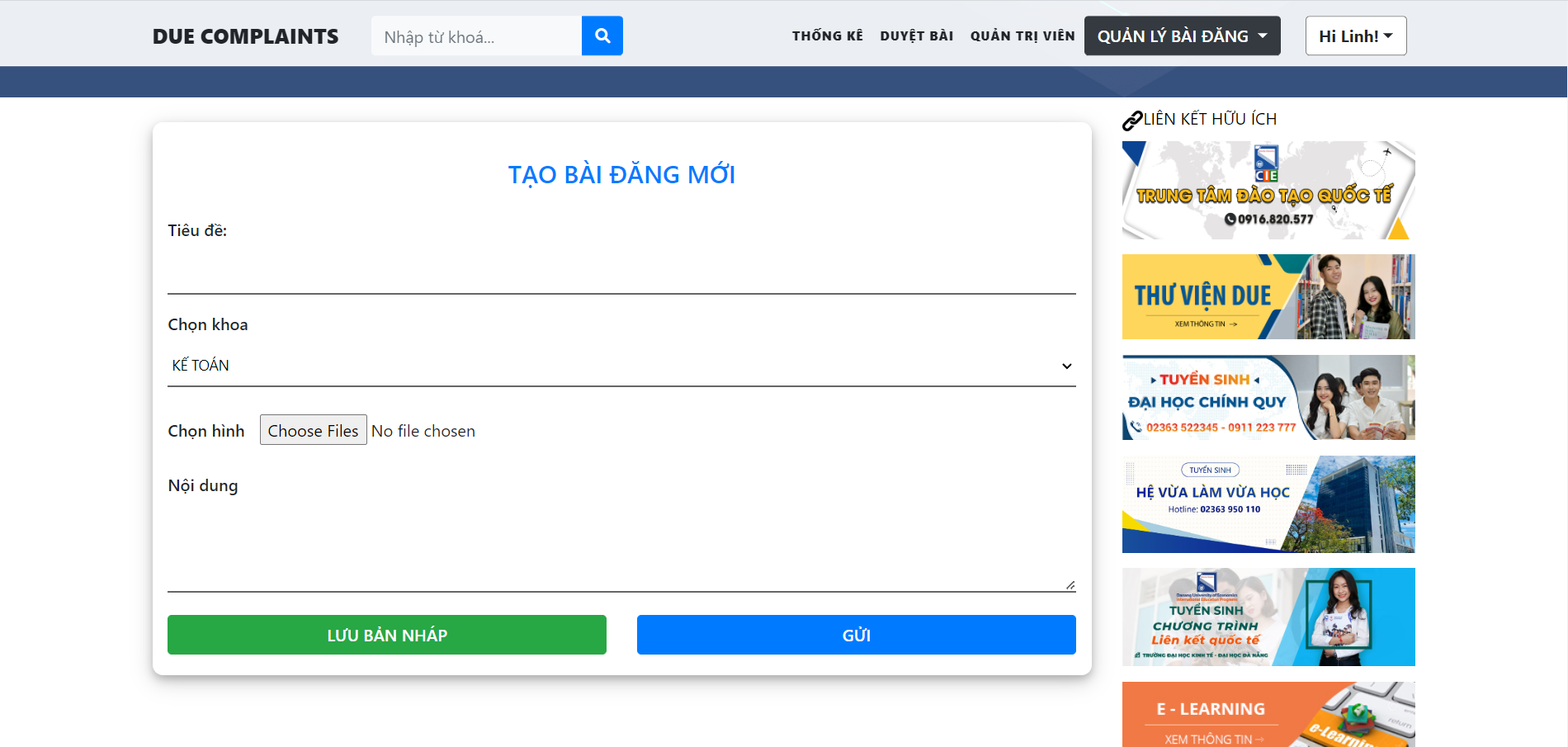
Hình 29. Giao diện trang thay đổi mật khẩu



Hình 30. Giao diện trang xem bài đăng chi tiết

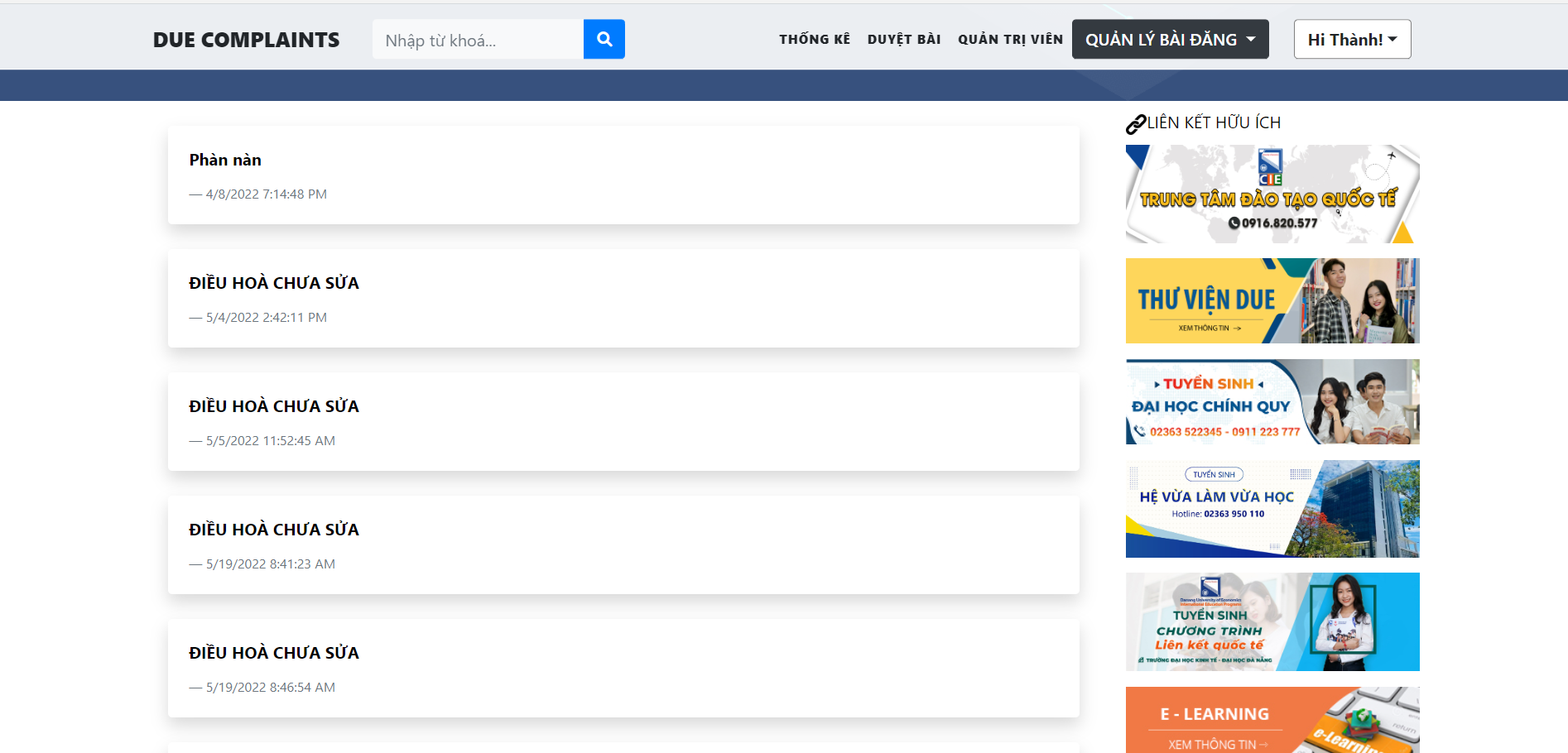


Hình 31. Giao diện trang xem bài đăng tổng quan cá nhân

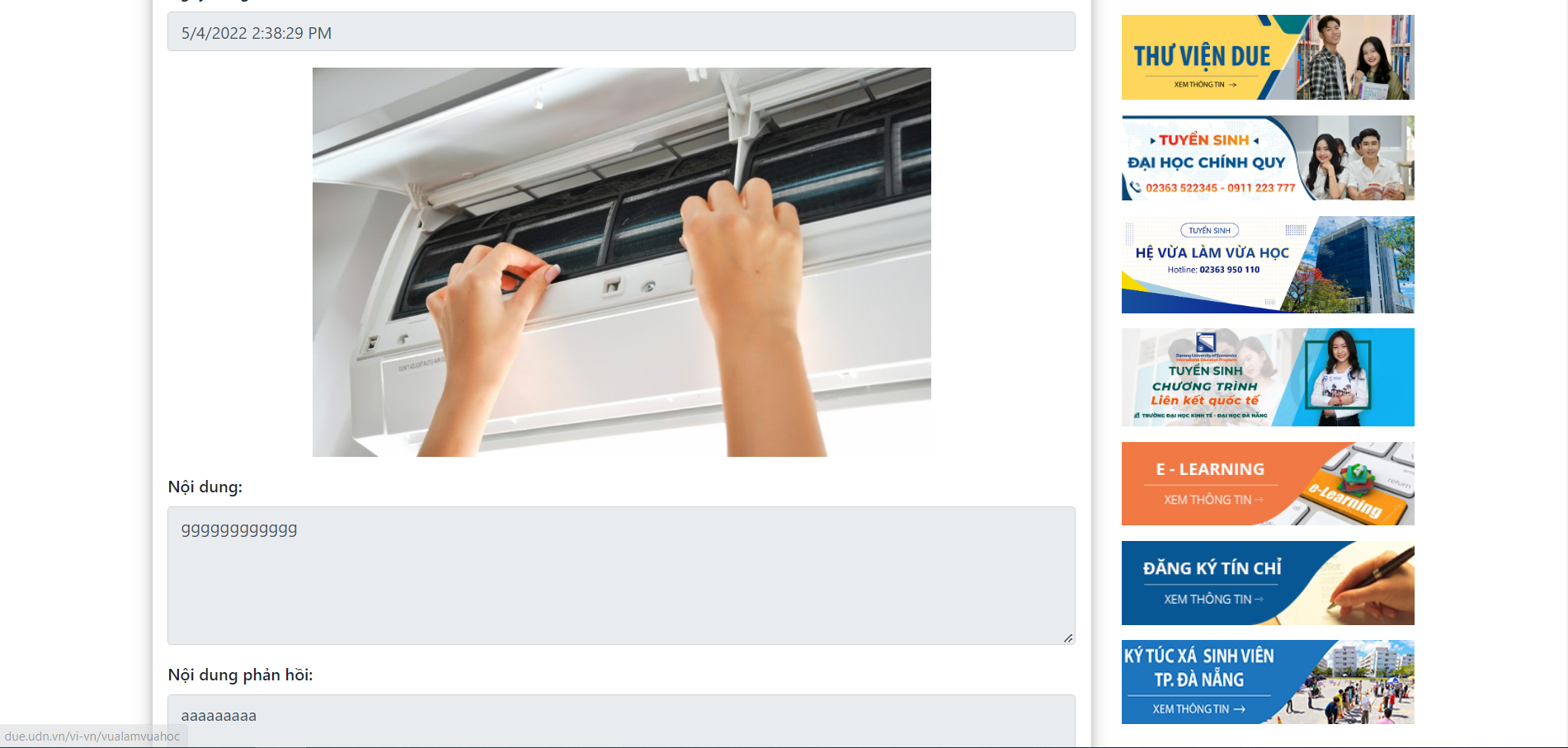


Hình 32. Giao diện trang tạo bài đăng mới

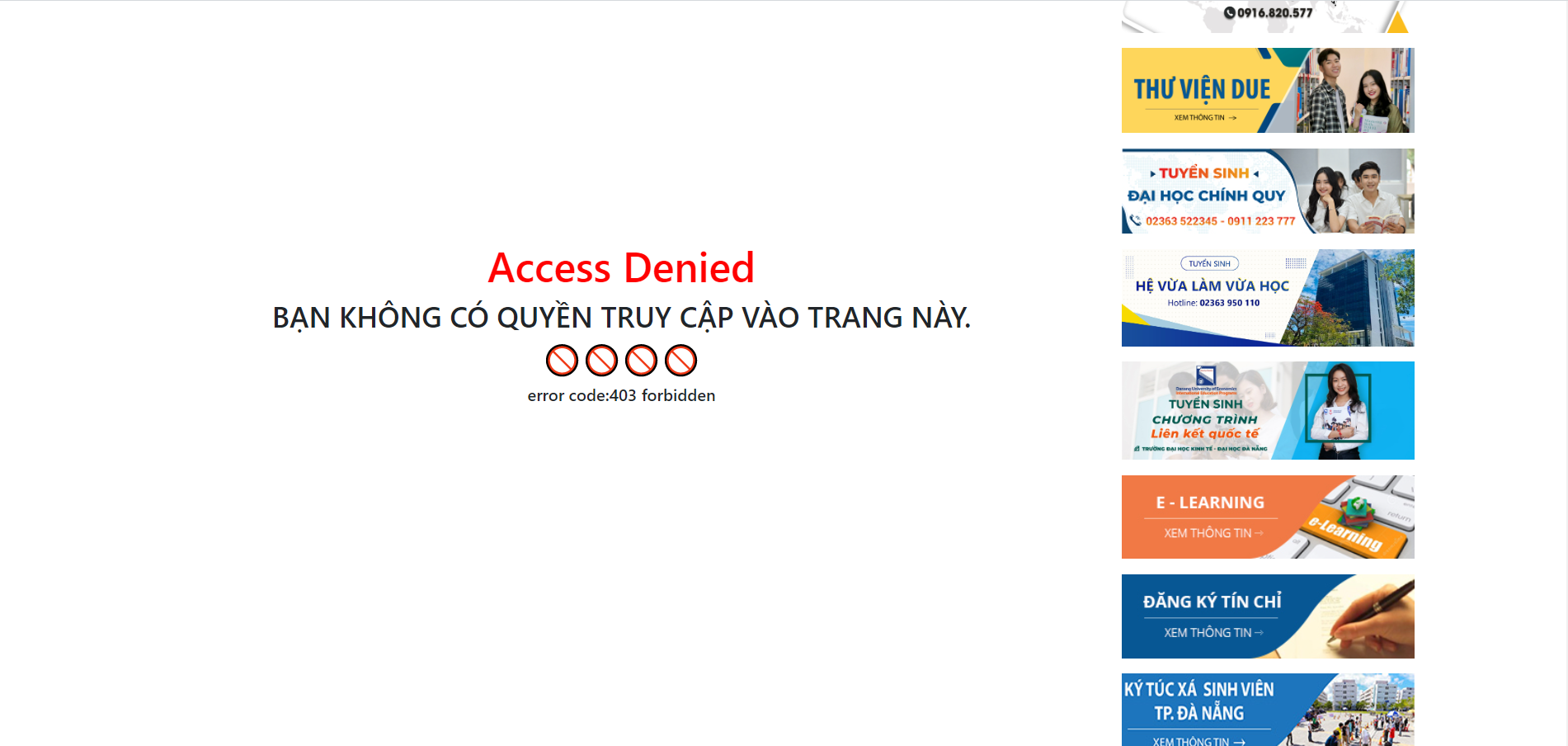
#### Các giao diện chính của nhân viên phòng ban:



Hình 33. Giao diện trang duyệt bài đăng tổng quan( Nhân viên)

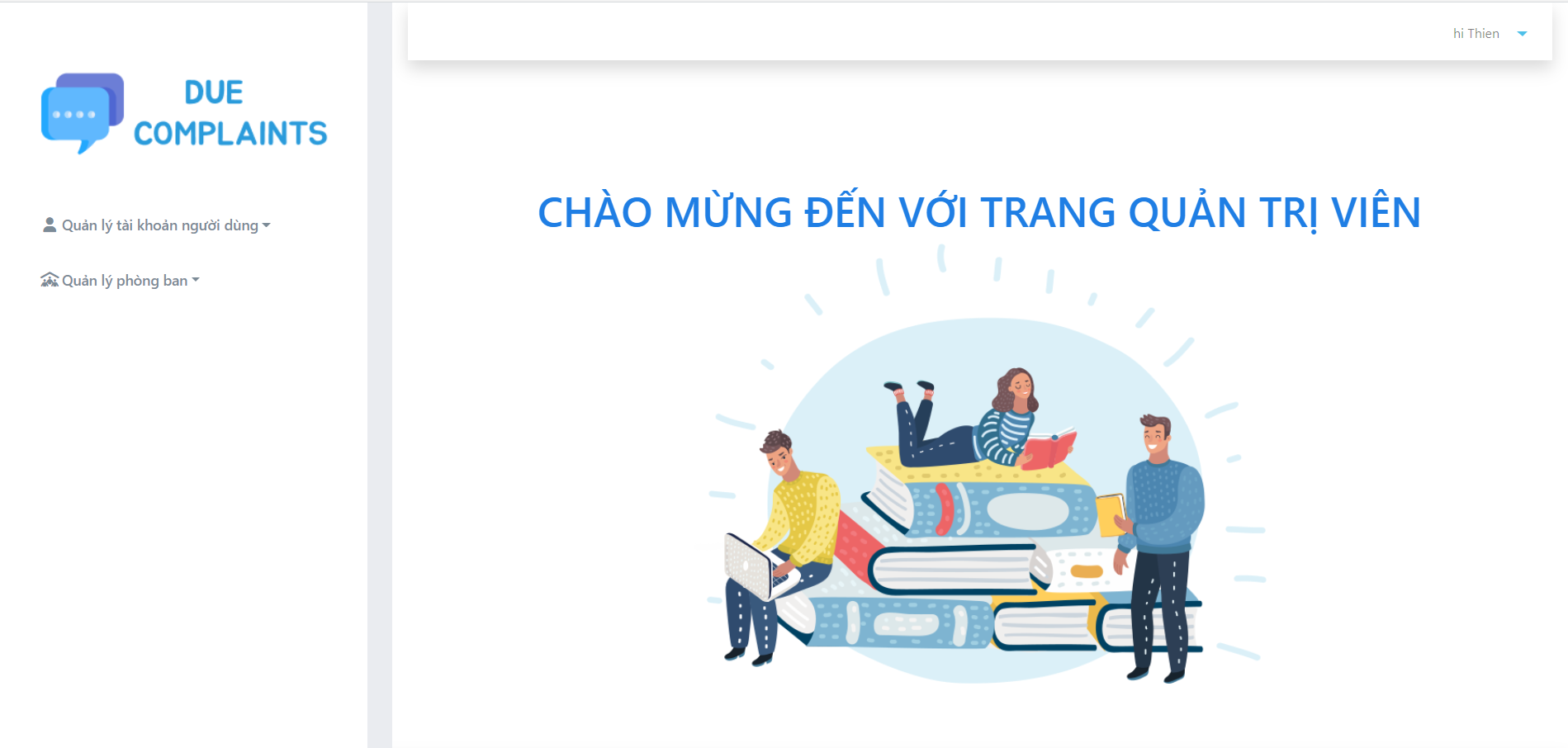


Hình 34. Giao điện trang duyệt bài đăng chi tiết ( Nhân viên)

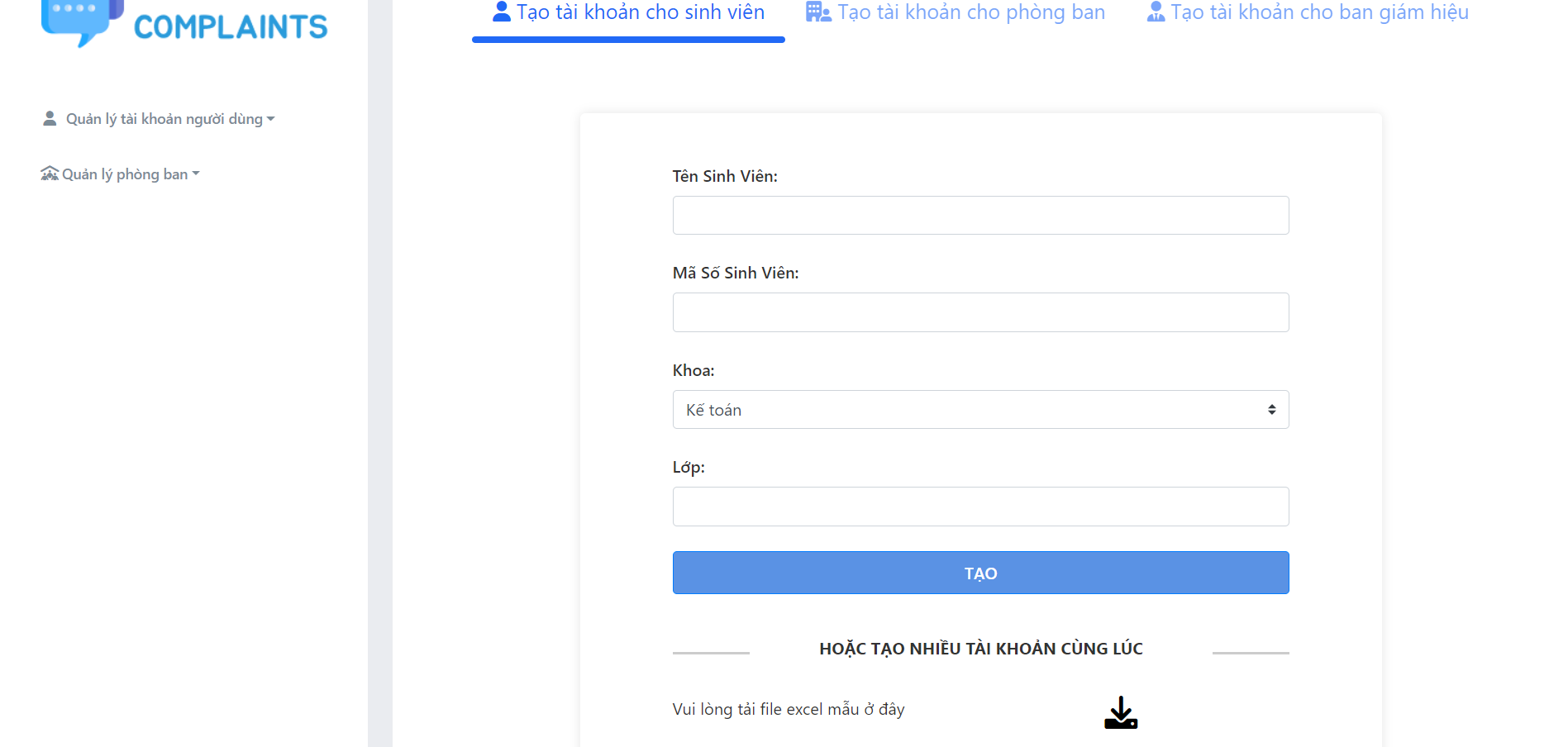


Hình 35. Giao diện trang Access Denied

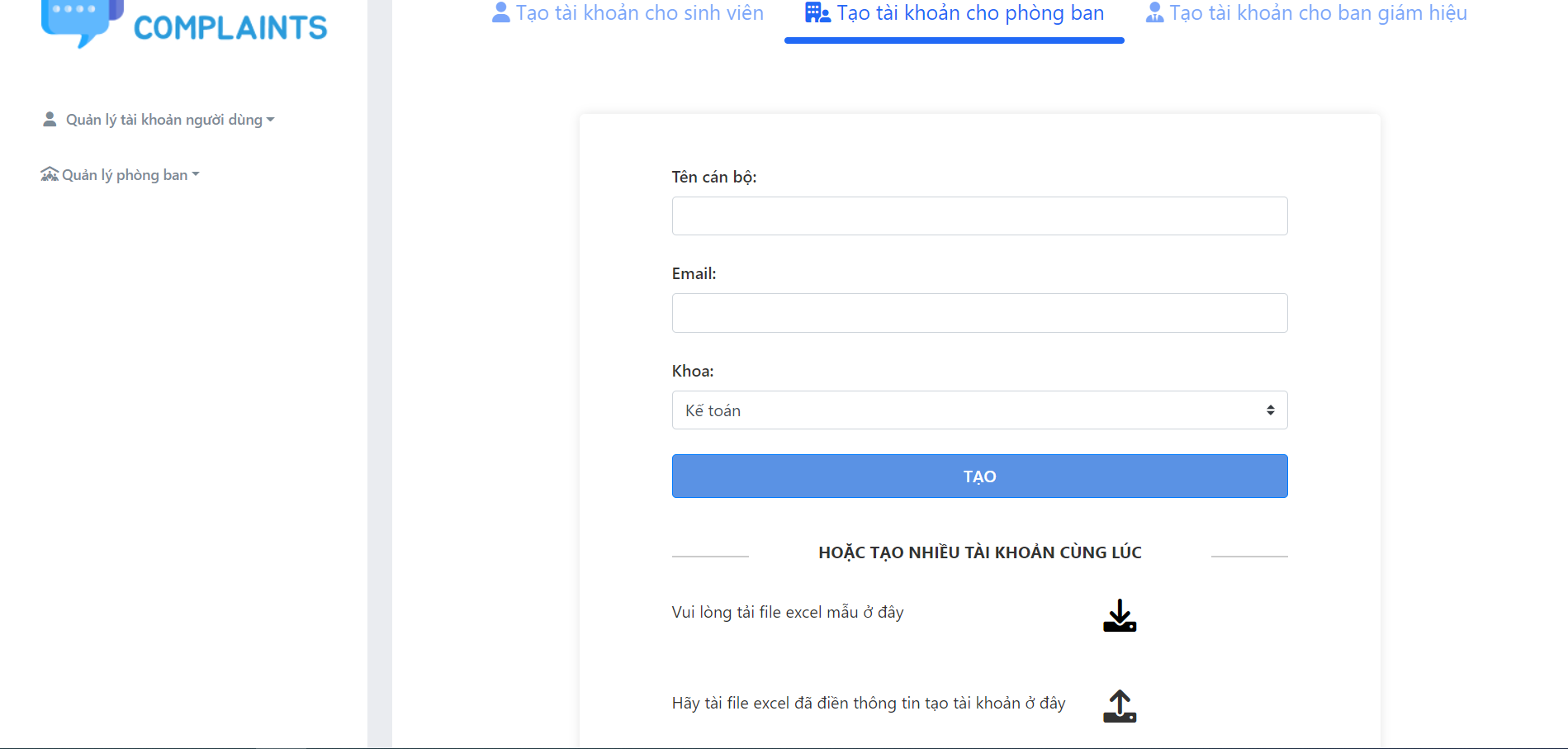
#### Các giao diện chính của quản trị viên:



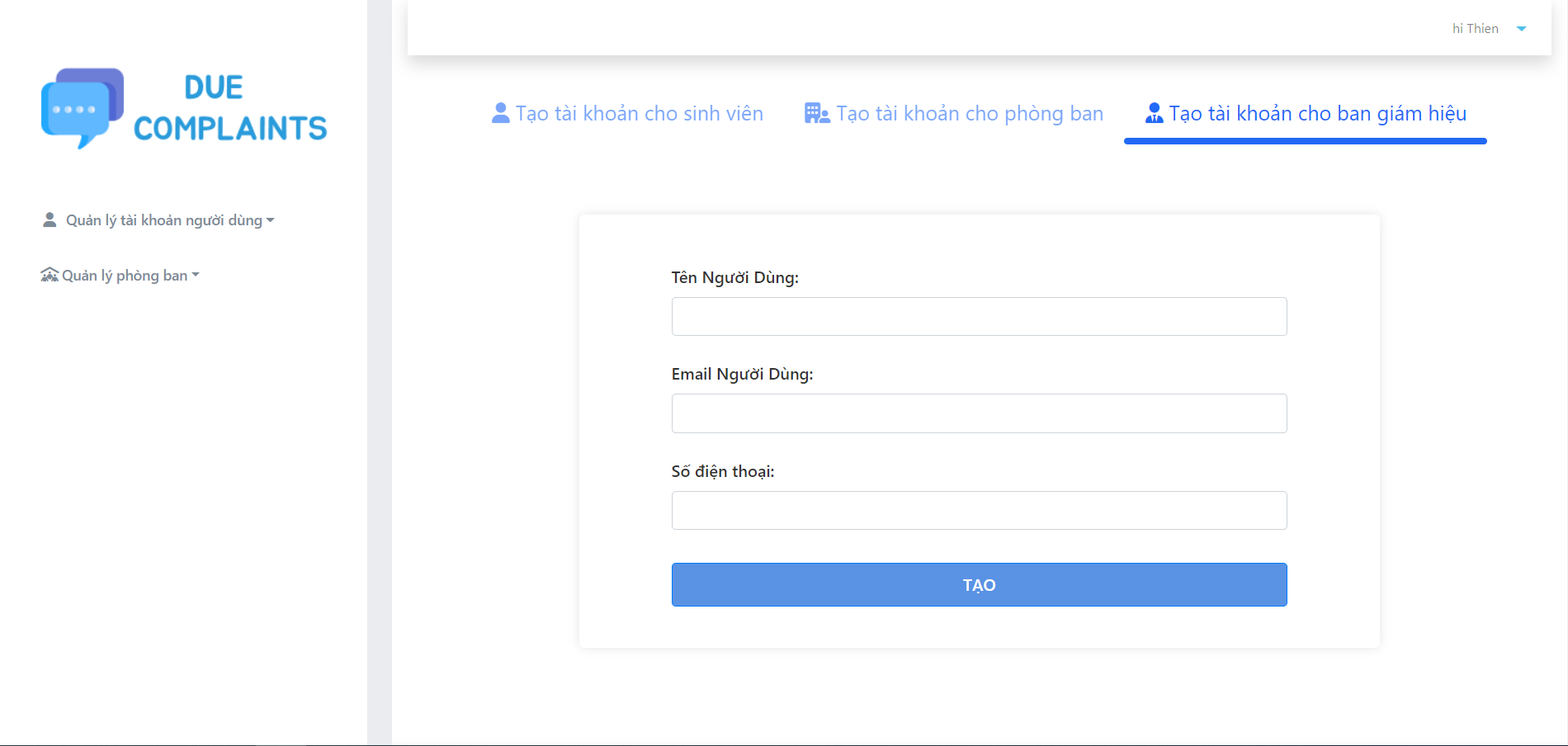
Hình 36. Giao diện trang chủ ( Quản trị viện)



Hình 37. Giao diện trang tạo tải khoản sinh viên



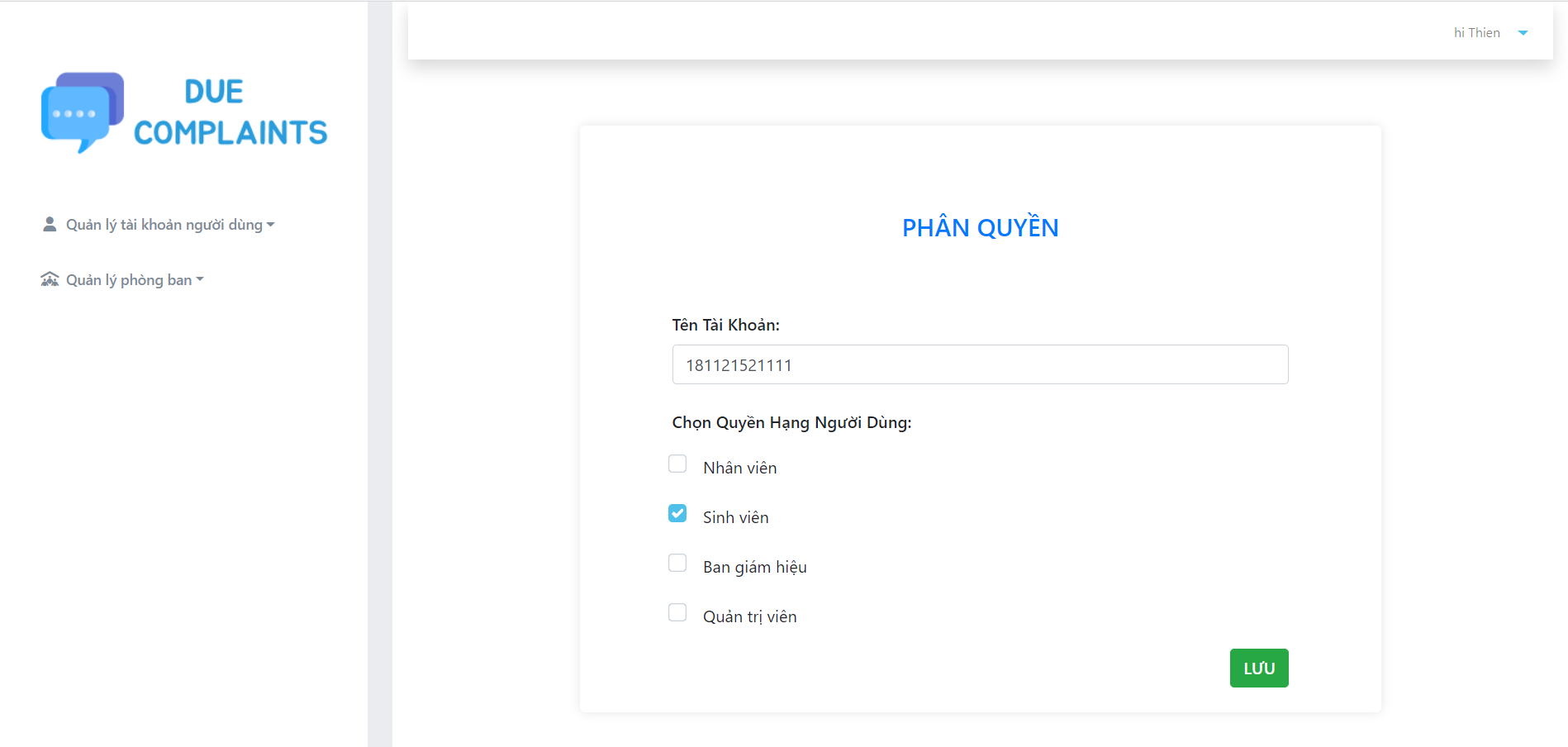
Hình 38. Giao diện trang tạo tải khoản cho phòng ban



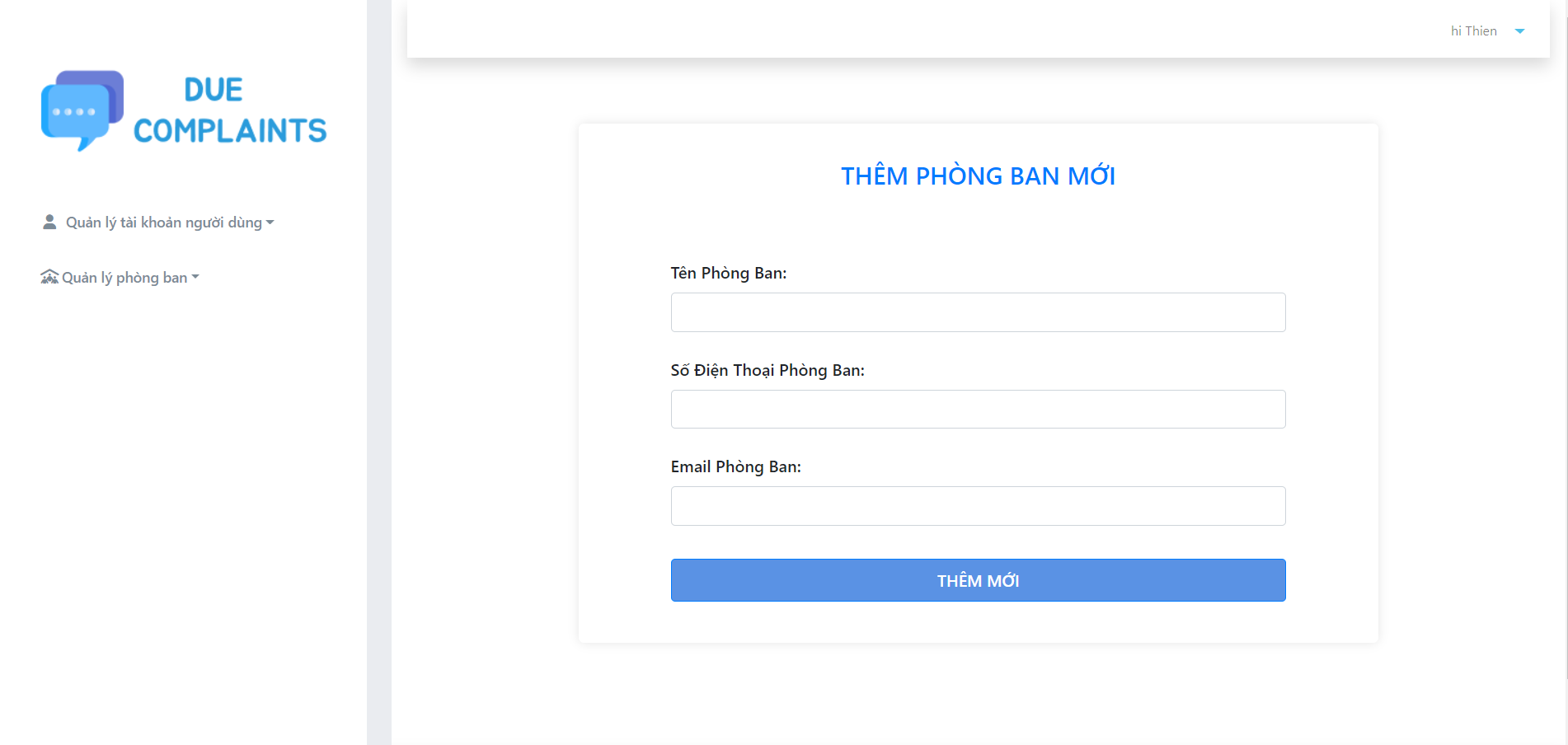
Hình 39. Giao diện trang tạo tài khoản ban giám hiệu



Hình 40. Giao diện trang quản lý tài khoản người dùng



Hình 41. Giao diện trang phân quyền

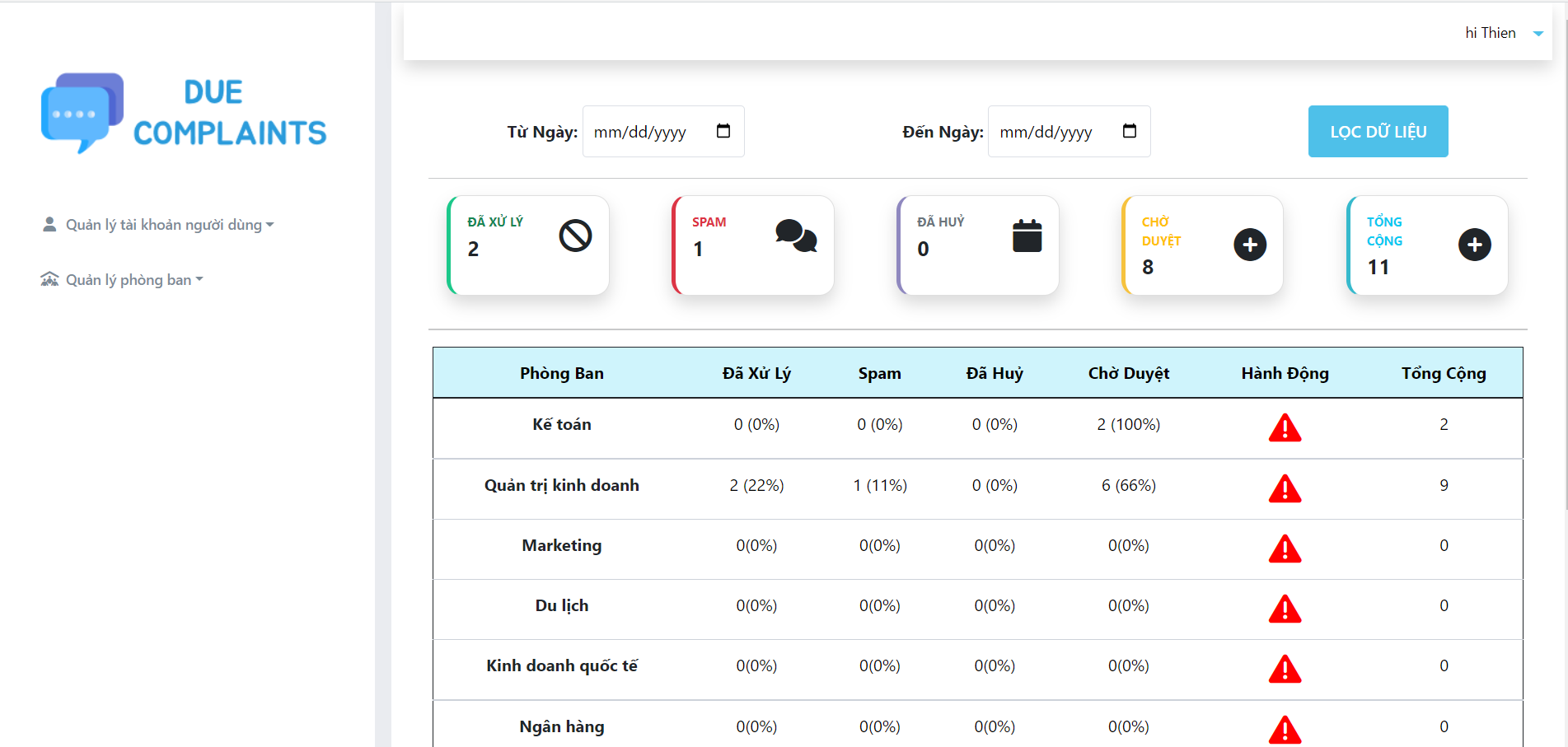


Hình 42. Giao diện trang tạo mới phòng ban



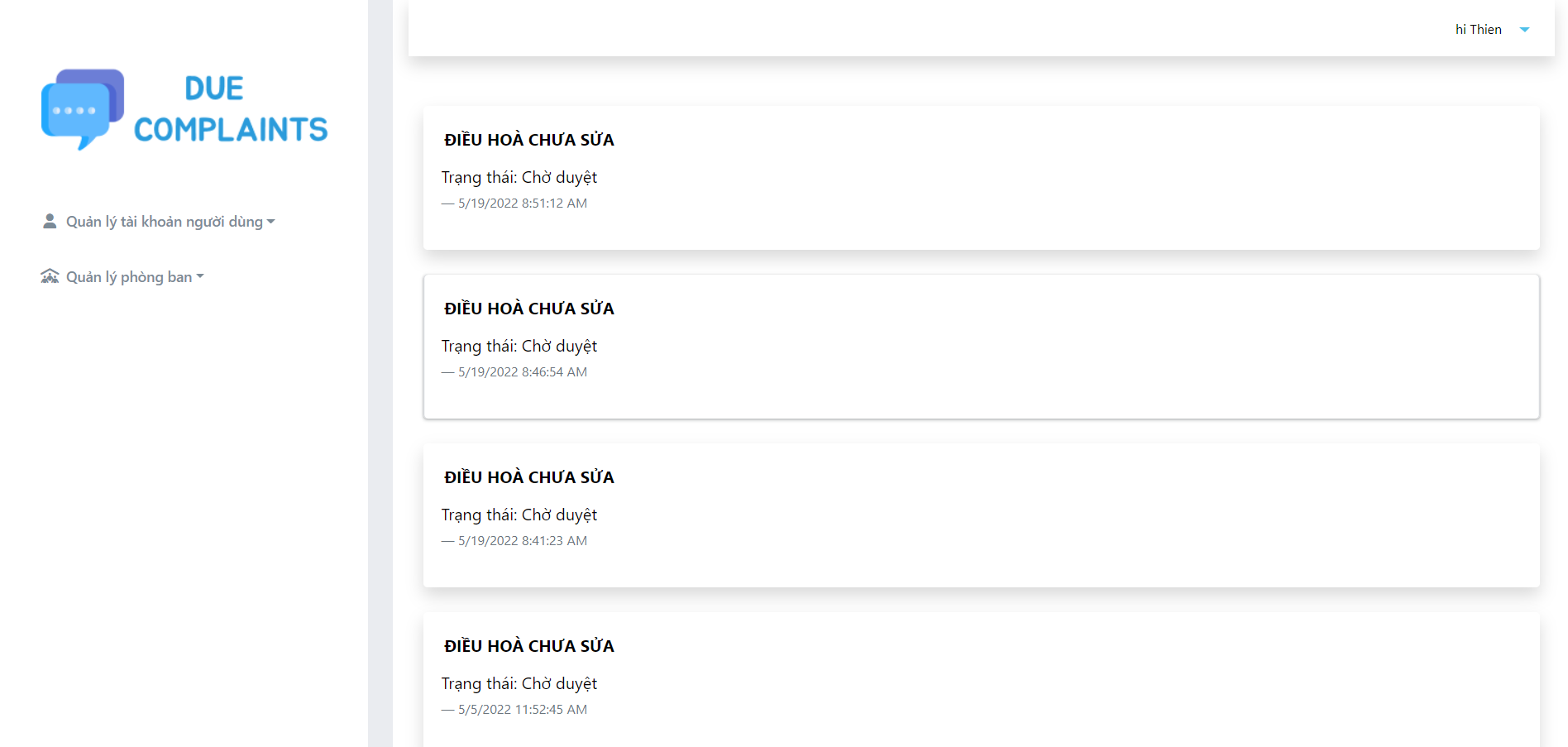
Hình 43. Giao diện trang quản lý phòng ban

#### Giao diện thống kê của ban giám hiệu:



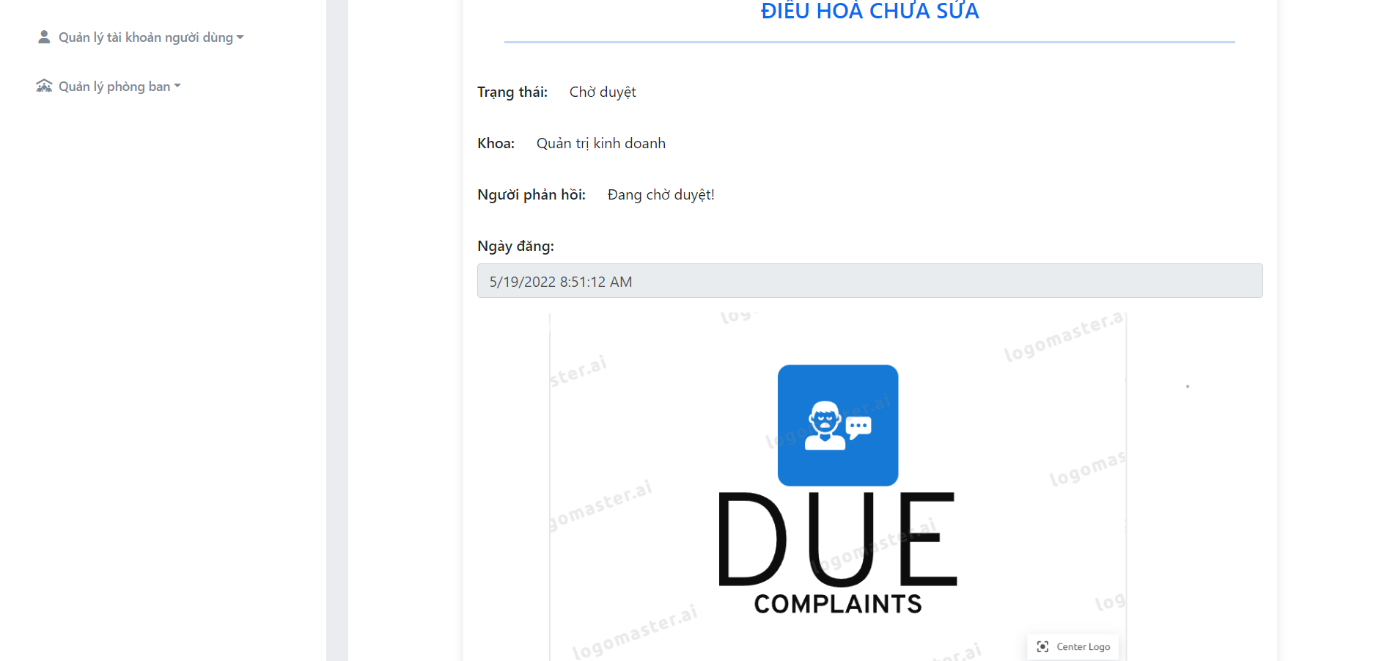
Hình 44. Giao diện trang thống kê

#### Giao diện xem danh sách bài đăng theo từng loại của ban giám hiệu:



Hình 45. Giao diện xem danh sách bài đăng theo loại của ban giám hiệu

#### Giao diện xem bài đăng chi tiết của ban giám hiệu:



Hình 46. Giao diện xem chi tiết bài đăng của ban giám hiệu

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## Kết luận

### Đánh giá tổng quan kì thực tập

Trải qua kì thực tập tốt nghiệp tại công ty BYS đã giúp em mở mang thêm nhiều kiến thức mới cũng như các kinh nghiệm thực tiễn. Mở ra cho em những bước đầu trên con đường trở thành một lập trình viên chuyên nghiệp:

* Được làm việc và học tập trong một môi trường chuyên nghiệp cùng với các anh chị, từ đó thúc đẩy sự tìm tòi, phát triển bản thân hoàn thiện hơn về mặt kiến thức cũng như kĩ năng, nâng cao khả năng giao tiếp, hoà nhập văn hoá công ty.
* Trong khoảng thời gian thực tập cũng sẽ gặp nhiều khó khăn vì cần kiến thức chuyên môn cao, nhưng cũng vì vậy đã khiến em có cơ hội được thử thách bản thân mở rộng giới hạn hơn, bên cạnh đó cũng học hỏi và nghiên cứu được nhiều kiến thức mới.

### Đánh giá về project trong kì thực tập

Dự án *“Xây dựng Website Tiếp nhận phản ánh Trường Đại học Kinh Tế - Đại học Đà Nẵng”* được xây dựng hướng tới 2 đối tượng chính là sinh viên và đối tác như phụ huynh, doanh nghiệp, … nhằm mục đích tạo một môi trường giáo dục lành mạnh và chuyên nghiệp. Nhưng vì thời gian và nguồn lực có hạn nên hiện tại chỉ tập trung vào đối tượng chính là sinh viên nhằm hoàn thiện một cách tuyệt đối các chức năng lớn đến những phần nhỏ nhất của hệ thống, đưa ra một sản phẩm hoàn chỉnh.

Hệ thống có đầy đủ các chức năng cơ bản cũng như các chức năng nghiệp vụ quản trị dành cho admin, có thể quản lý người dùng, quản lý các phòng ban. Ngoài ra, hệ thống cũng có một số tính năng để tạo cảm giác thân thiện với người dùng hơn, như là gửi mail thông báo về trạng thái bài viết, tạo tài khoản hàng loạt bằng file Excel, …

### Experience trong kì thực tập

Trải qua 6 tháng thực tập tại BYS ngoài những lợi ích về vật chất còn cho em rút ra được những bài học, kinh nghiệm mà có lẽ không buổi học nào ở trường có thể cho em được.

Đầu tiên là về sự chủ động trong công việc, đối với văn hoá tại BYS, mỗi người đều phải chủ động tự hoàn thành công việc của bản thân, không anh chị nào có thời gian để dìu dắt chúng em từng bước, từ đó đã rèn luyện cho em tính chủ động trong công việc. Tự tìm hiểu, tự học hỏi, tự nghiên cứu, chủ động hỏi mentor khi có những vấn đề không thể tự giải quyết.

Tiếp theo là về cách ứng xử, đi làm được gặp nhiều đồng nghiệp hơn, nhiều đối tác hơn, tiếp xúc với nhiều người hơn, từ đó em học được cách ứng xử đối với từng người, biết tiết chế thái độ và không thể hiện bản thân quá mức.

Bên cạnh những kĩ năng mềm thì BYS cũng cho em học được nhiều kiến thức, kỹ năng chuyên môn vô cùng giá trị như là: kĩ năng lập trình logic hơn, xử lý vấn đề khi phát sinh một cách chuyên nghiệp và nhanh chóng, xây dựng một sản phẩm theo quy trình chuyên nghiệp và logic, … có thể em sẽ được học những điều này từ sách vở nhưng sẽ mất rất nhiều thời gian, thời gian thực tập tại BYS đã giúp em đẩy nhanh quá trình trở thành một lập trình viên chuyên nghiệp.

## Hướng phát triễn sau dự án

### Hướng phát triển của bản thân

Qua 6 tháng thực tập tại BYS, em nhận thấy được bản thân vẫn còn nhiều thiếu xót trong công việc cũng như là kiến thức về lập trình, nhưng cũng cho em nhìn thấy được rõ ràng vị trí của bản thân và xác định được những bước đi trong tương lai trên con đường trở thành lập trình viên chuyên nghiệp.

Để tiếp tục phát triển bản thân trên con đường lập trình viên chuyên nghiệp, em phải tiếp tục học thêm nhiều kiến thức và không ngừng nỗ lực để bắt kịp xu hướng thời đại công nghệ mới.

Định hướng lập trình sẽ theo nghiên cứu trong tương lai:

* Front - end: Agular.
* Back - end: DotNet.

### Hướng phát triển của dự án

Vì nguyên nhân thời gian cũng như nguồn lực, vẫn chưa thể hoàn thiện tuyệt đối và còn nhiều điểm chưa hoàn hảo nhưng dự án cũng mang ý nghĩa thực tế và có thể ứng dụng vào đời thực. Trong tương lai sẽ phát triển thêm dự án với những khía cạnh:

* Xây dựng hệ thống phần của đối tác, phụ huynh.
* Nâng cấp và làm mịn giao diện hệ thống để website đẹp hơn và thân thiện với người dùng.
* Xây dựng các tính năng bổ trợ như: box chat, thông báo, báo cáo bài viết, …

TÀI LIỆU THAM KHẢO

1. **Codegym.vn**
2. [**https://tuhocict.com/**](https://tuhocict.com/)